

Ontwikkeling advertentiewebsite

Afstudeerproject VIA

Afstudeer Periode	9 februari 2004 / 11 juni 2004
Naam	Marc Schut
Opleiding	VIA
Bedrijf	Damman Software
Datum	11-06-04
Examinatoren:	J.P van Leeuwen W. Nijenhuis

Colofon

Project gegevens

Studentnummer: 99004023

Module: Afstuderen

Examinatoren: W. Nijenhuis, J.P. van Leeuwen

Opleidinggegevens

Informatica (Vormgeving en Ontwerp van Interactie)

Haagse Hogeschool

Sector Informatica

Johanna Westerdijkplein 75

2521 EN Den Haag

Opdrachtverlenend bedrijf

Damman Software

Ambachtsweg 18

8152 BA Lemelerveld

Referaat

In dit verslag wordt het proces beschreven om een advertentiewebsite te maken die Marc Schut heeft uitgevoerd in de periode van 9 februari 2004 t/m 11 juni 2004.

Descriptoren

- Alan Cooper
- Website
- Java
- Modelleren

Voorwoord

Om af te studeren voor de opleiding Vormgeving en Ontwerp van Interactie (VIA) op de Haagse Hogeschool (HHS) moest er een afstudeerproject worden doorlopen. In dit project wordt er geacht dat de theoretische onderdelen van de opleiding in de praktijk worden uitgevoerd met een afstudeeropdracht.

Een geschikte opdracht vinden voor het afstuderen is dan ook één van de belangrijkste punten van het afstuderen.

Mijn dank gaat uit naar de examinatoren W. Nijenhuis, J.P. Van Leeuwen en eigenaar van Damman Software, Roy Damman. Dankzij de interesse van deze personen in deze opdracht heb ik de kans gekregen om mijn geleerde kennis in de praktijk uit te voeren, alsmede mijn kennis te verbreden in een nieuwe programmeertaal.

Raalte, 22-05-04

Marc Schut

Voorwoord.....	4
1 Inleiding.....	7
2 Organisatie.....	9
2.1 Algemene beschrijving van de organisatie.....	9
2.2 Plaats van de student in de organisatie.....	9
3 De afstudeeropdracht.....	10
4 Oriëntatiefase.....	12
4.1 Oriënteren op Linux.....	12
4.1.1 Linux installeren als webserver.....	13
4.1.2 Linux instellen als ontwikkelomgeving.....	13
4.2 Oriënteren op Java.....	14
4.3 Installatie methode en instellingen vastleggen.....	14
4.4 Usability onderzoek vergelijkbare websites.....	14
5 Analysefase.....	19
5.1 Opstellen plan van aanpak.....	19
5.1.1 Keuze van methode voor de opdracht.....	19
5.1.2 Onderverdelen van het ontwikkelproces in fases.....	21
5.1.3 Doelen opstellen van de opdracht.....	22
5.1.4 Opstellen van een globale planning.....	23
5.2 Opstellen definitiestudie.....	23
5.2.1 Persona's opstellen.....	23
5.2.2 Doelen opstellen aan de hand van persona's	25
5.2.3 Eisen en wensen opstellen van de gebruikers.....	25
5.2.4 Functionele omschrijving opstellen.....	27
5.2.5 Website opsplitsen in pilots.....	28
5.2.6 Verificatie met de opdrachtgever	30
6 Conceptfase.....	31
6.1 Keuze maken van geschikte modellen.....	31
6.2 Modelleren van Use-casediagram.....	33
6.3 Modelleren van Classdiagram.....	35
6.4 Modellen verifiëren met Damman Software.....	36
7 Ontwikkelingsfase.....	37
7.1 Uitwerken Pilot 1 Database installeren en opzetten.....	37
7.2 Uitwerken Pilot 2 Uitwerken van broncode voor functionaliteit van de website.....	38
7.2.1 Opzetten en programmeren van de structuur van de broncode.....	38
7.2.2 Programmeren van de functionaliteit van de website.....	40
7.3 Uitwerken Pilot 3: Uitwerken van de code voor de pagina's van de website.....	43
8 Testfase.....	46
8.1 Testen van de website op bugs.....	46
8.1.1 Testen van website met Black Box Testing.....	47
8.1.2 Testen van website met White Box Testing.....	49
8.2 Testen van de website aan de hand van persona's.....	50
8.3 Opstellen van een wijzigingsrapport.....	51
9 Evaluatie van de afstudeeropdracht.....	52
9.1 Evaluatie van het gevolgde proces.....	52
9.1.1 Evaluatie van de gebruikte methode.....	52
9.1.2 Evaluatie van de gebruikte technieken.....	53

9.2 Evaluatie van de website	53
Woordenlijst.....	54
Bronnen.....	56
Bijlage A: Volledige opdrachtsomschrijving.....	58
Bijlage B: Resultaten Usability onderzoek vergelijkbare websites.....	62
Bijlage C: Globale planning.....	67
Bijlage D: Systeemeisen.....	69
Bijlage E: Onderzoek aanvaardbaarheid systeemeis.....	72
Bijlage F: Representative qualities	74
Bijlage G: UML Klassendiagram.....	79
Bijlage H: UML Use-Case diagram.....	80
Bijlage I: Planning uitwerken van pilots.....	81
Bijlage J: Programeerrichtlijnen.....	84

Om af te studeren bij de opleiding VIA moet er aan het einde van de opleiding 100 dagen lang een opdracht worden uitgevoerd bij een bedrijf. Na het uitwerken van de opdracht wordt deze gekeurd aan de hand van dit document.

In dit verslag wordt een complete beschrijving gegeven van de werkzaamheden van het project. Hierbij wordt niet alleen een beschrijving gegeven van de activiteiten, maar ook worden afwegingen, gemaakte keuzen en oplossingen beschreven. Het doel van het verslag is om de examinatoren voldoende inzicht te geven in:

- De omvang van de opdracht;
- De diepgang waarmee verschillende aspecten zijn uitgewerkt;
- De wijze waarop producten tot stand zijn gekomen.

Het verslag is als volgt opgebouwd:

Hoofdstuk 2

In hoofdstuk 2 wordt een korte omschrijving gegeven van het bedrijf, de organisatie waar de afstudeeropdracht is volbracht.

Hoofdstuk 3

In hoofdstuk 3 wordt de opdracht omschreven die uitgevoerd is voor het afstuderen. Deze opdracht is de exacte omschrijving zoals deze is gekeurd door de examinatoren.

Hoofdstuk 4

In dit hoofdstuk wordt de beginperiode van het afstuderen belicht. Het gaat hierbij om de kennis die is opgedaan bij het oriënteren naar de programmeertaal en het operating systeem wat gebruikt gaat worden en een onderzoek naar vergelijkbare producten.

Hoofdstuk 5

In hoofdstuk 5 wordt de eerste echte fase van het project, de analysefase beschreven. In de analysefase worden de keuzes toegelicht die gemaakt zijn bij het opstellen van een plan van aanpak en het definiëren van de specificaties voor de opdracht.

Hoofdstuk 6

In hoofdstuk 6 wordt de opdracht gemodelleerd, dit wil zeggen dat de technische structuur van de opdracht wordt weergegeven door verschillende modellen. Dit hoofdstuk geeft inzicht in de modellen en de gemaakte keuzes hierbij.

Hoofdstuk 7

Hoofdstuk 7 is de ontwikkelingsfase. Dit hoofdstuk gaat over het uitwerken van het product, wat voldoet aan de specificaties die opgesteld zijn in de analysefase en in de conceptfase. Hierbij zullen enkele keuzes worden uitgelegd en de stappen beschreven die hiervoor zijn gedaan.

Hoofdstuk 8

In hoofdstuk 8 wordt de laatste fase beschreven van het project. In de testfase wordt

uitgelegd welke tests er zijn uitgevoerd om het product te testen aan de gestelde eisen die opgesteld zijn. Ook de resultaten van deze tests zullen besproken worden.

Aan het eind van dit verslag vindt u de bronnenlijst, een lijst met begrippen en interne bijlagen.

2.1 Algemene beschrijving van de organisatie

Damman Software is een eenmansbedrijf dat is opgericht in 1992 door Roy Damman. Nadat het bedrijf in de eerste jaren in Wijhe was gevestigd is Damman Software in 1999 verhuist naar Lemelerveld.

Op deze locatie ontwerpt, programmeert en verkoopt men software op maat voor het midden en kleinbedrijf. De software biedt vaak een optimalisatie en automatiseren van het werkproces bij logistieke systemen.

2.2 Plaats van de student in de organisatie

De student heeft de mogelijkheid gekregen om deze afstudeeropdracht uit te voeren en te volbrengen bij Damman Software. De student wordt gezien als een software ontwikkelaar in opleiding. Damman Software zorgt voor goede ondersteuning bij het opzetten en programmeren van een systeem.

Hieronder volgt een korte omschrijving van de afstudeeropdracht die is goedgekeurd door de Haagse Hogeschool bij aanvang van het afstuderen. Zie Bijlage A: Volledige opdrachtsomschrijving voor de volledige omschrijving.

Momenteel zijn er veel websites gericht op het aanbieden van tweedehands goederen. Deze websites zijn vaak niet consistent qua opzet en bieden niet de gewenste zoekmogelijkheden. Het is de bedoeling een advertentiewebsite te maken die deze Websites overtreft in het gebruiks- en zoekgemak.

Doordat er gebruik gemaakt wordt van kenmerken per advertentiecategorie kan er via een uitgebreide zoekfunctie gericht worden gezocht dan bij de huidige advertentiewebsites. Damman Software denkt dat dit gemak extra bezoekers, adverteerders en bedrijven aan zal trekken. Hierdoor kunnen er tarieven gevraagd worden voor advertenties, reclame en aanvullende diensten.

Daar het beoogde aantal bezoekers groot is, moet getracht worden het dataverkeer zo beperkt mogelijk te houden. Dit moet worden bereikt zonder dat dit ten koste gaat van de gebruikerservaring.

De beheerder zal via een beheerdersgedeelte de site kunnen instellen en de advertenties en gebruikers kunnen beheren. Om ervoor te zorgen dat er geen ongewenste advertenties worden geplaatst moet er een manier worden gevonden om dit tegen te gaan. Op dit moment wordt er gedacht aan een bevestigingssysteem via email.

Het doel van de website is het genereren van inkomsten voor Damman Software. Hiervoor moet de website worden voorbereid op het invoeren van commerciële toepassingen. Er wordt hierbij gedacht aan het betalen voor advertenties, reclame en aanvullende diensten. In eerste instantie ligt de nadruk op het aantrekken van zoveel mogelijk bezoekers en adverteerders. Hierdoor wordt het in een later stadium makkelijker om bedrijven aan te trekken en inkomsten te genereren.

Het doel van de afstudeeropdracht is het ontwikkelen van een advertentiewebsite. Deze website moet gericht kunnen zoeken op advertenties dan bestaande advertentiewebsites. Om inkomsten te kunnen genereren moet de website worden voorbereid op commerciële toepassingen. Een randvoorwaarde van de site is het beperken van onnodig dataverkeer en de belasting van de server.

Door de ontwikkelomgeving gelijk te stellen aan de productie-omgeving van de site zullen de opstartproblemen zo beperkt mogelijk worden gehouden. Dit houdt in dat de Website wordt ontwikkeld op basis van Java Servlets, Java Server Pages draaiend op een Linux, Apache machine. Als Relationeel DataBase Management Systeem wordt gebruik gemaakt van Firebird. De keus voor Firebird is gemaakt aan de hand van goede ervaringen van Damman Software. Firebird is een veelzijdige en gratis database management systeem dat goed presteert met Linux als operating system.

Als ontwikkelprogramma wordt gebruik gemaakt van NetBeans. De domeinnaam

www.indemarkt.info is reeds geregistreerd en kan gebruikt worden om de site op zetten en te testen.

Bij Damman Software is voldoende kennis in huis om de opdracht goed te kunnen ondersteunen. Er zijn ook meerdere voorbeelden van andere reeds ontwikkelde sites die eventueel gebruikt kunnen worden voor deze opdracht.

De student heeft bij aanvang van de afstudeerperiode een basiskennis Java, geen kennis van Linux en NetBeans. De student heeft wel veel ervaring in het ontwikkelen van Websites met een database koppeling.

Als voorbereiding op de afstudeeropdracht is een oriëntatiefase een geschikte fase om kennis te maken met bijvoorbeeld de programmeertaal of andere technische aspecten van de opdracht. Door deze kennismaking wordt het makkelijker om eisen beter te beoordelen op haalbaarheid en benodigde tijd in te schatten voor het uitwerken ervan.

Aangezien er door Damman Software al enige technische standaarden gegeven waren, zoals het programmeren in Java en het werken met Linux zijn deze punten in de oriëntatiefase al verkend.

Een belangrijke vernieuwing ten opzichte van de huidige advertentiesites is het bieden van meer mogelijkheden om een advertentie te vinden en te plaatsen. De bedoeling is deze mogelijkheden de gebruikersvriendelijkheid van de website zullen verhogen. Om erachter te komen wat nu de positieve en negatieve kanten zijn van de huidige advertentiewebsites is er gekozen om een vooronderzoek op usability van concurrerende websites op te nemen in de oriëntatiefase.

4.1 Oriënteren op Linux

Linux is een besturingssysteem voor computers en kan worden gebruikt als alternatief voor bijvoorbeeld Windows. Linux wordt aangeboden in de vorm van distributies. Een distributie bevat naast het besturingssysteem zelf ook een groot aantal applicaties. Hiermee worden de normale programma's zoals tekstverwerkers, rekenprogramma's, emailprogramma's en webbrowsers bedoeld.

Enkele samenstellers van distributies zijn Red Hat, Suse, Mandrake, Knoppix en vele andere. Het grote voordeel van Linux is dat de broncode (bouwtekening) van Linux vrij beschikbaar is voor iedereen. Een groot nadeel is dat niet alle softwarefabrikanten hun programma's beschikbaar hebben gemaakt voor Linux. Ook de gebruikersvriendelijkheid voor computerbeginners laat nog wat te wensen over. Er wordt echter op dit vlak in een hoog tempo vorderingen geboekt.

Damman Software heeft voor Linux gekozen omdat het stabiel werkt en veilig en Open Source is. Een goed Linux systeem draait zonder probleem honderden dagen zonder een herstart.

Bij aanvang van deze fase was de kennis van Linux zeer beperkt. De kennis over de mogelijkheden was aanwezig, maar ik had nog nooit gewerkt met Linux als besturingssysteem. Damman Software had speciaal voor deze oriëntatie een pc beschikbaar gesteld die als server moest gaan dienen om de advertentiewebsite op de hosten. Het workstation wat als ontwikkelomgeving gebruikt zou worden was al ingericht met Linux. Hierbij moesten echter ook een aantal aanpassingen gedaan worden.

4.1.1 Linux installeren als webserver

Damman Software gaat de advertentiewebsite draaien op Linux. Bij aanvang van de afstudeerperiode had Damman Software een nieuwe pc gekocht die als webserver zou dienen. Door mij de installatie en configuratie van Linux te laten doen, was zowel de webserver klaar om gebruikt te worden en had ik kennis gemaakt met de beginselen van Linux. Damman Software had al kennis van Linux en heeft mij dan ook ondersteund bij de installatie hiervan.

Het installeren van Linux was snel gebeurd, daar Damman Software een volledige distributie van Linux op CDRom had staan. Bij het installeren van Linux werd de hardeschijf geformatteerd naar een Linux bestandsindeling, waarna de benodigde bestanden gekopieerd werden.

Aangezien deze Linuxversie zou dienen als webserver moesten er nog een aantal andere programma's worden geïnstalleerd, zoals een databaseserver en een webserver. Linux heeft namelijk verschillende manieren voor het installeren van software. Ook het de-installeren van een programma verschilt nogal. Dit gebeurt niet altijd centraal en kan ook niet bij alle programma's. Dit is één van de tekortkomingen van Linux qua gebruikersvriendelijkheid.

Voor het installeren van een programma moet het desbetreffende programma eerst worden gedownload van het internet, waarna deze geïnstalleerd kan worden. Vergeleken met Windows betekent dit dat het niet altijd een kwestie is van dubbelklikken op een pictogram. Voor 80% van de applicaties moet een actie uitgevoerd worden in de console. Dit heeft veel weg van het MSDOS, wat in de jaren 80 veel werd gebruikt.

Na een kleine test met de testpagina's van de webserver is dit gedeelte van de oriëntatie afgesloten. Linux is geïnstalleerd en kan dienst doen als webserver.

4.1.2 Linux instellen als ontwikkelomgeving

Als tweede oefening om kennis te maken met Linux moest ik Linux als ontwikkelomgeving instellen. Dit betekende dat ik de website en de bijbehorende documenten moest maken met programma's die draaien op Linux. Op de hiervoor bestemde pc was al een distributie geïnstalleerd van Linux met OpenOffice. NetBeans IDE, Firebird databaseserver, J2SDK en Poseidon for UML moesten hier nog aan toegevoegd worden.

OpenOffice is een programma wat te vergelijken is met Microsoft Office. Deze is dan ook gebruikt om de documentatie mee te maken. Met NetBeans IDE is de website geprogrammeerd en getest op bugs (zie hoofdstuk 8.1 Testen van de website op bugs). Firebird databaseserver is gebruikt als RDBMS voor de website. De Java SDK (J2SDK) was noodzakelijk om NetBeans IDE en Poseidon Uml te gebruiken. Deze zijn namelijk geprogrammeerd in java. Net als bij het installeren van de programma's bij de vorige paragraaf zijn de programma's op verschillende wijze geïnstalleerd.

4.2 Oriënteren op Java

Java is de programmeertaal waar de website zou worden geprogrammeerd. Aangezien ik voor aanvang van dit project weinig tot geen ervaring had met Java heb ik eerst een aantal kleine functies geschreven in Java. Dit heb ik gedaan met NetBeans IDE om ook de mogelijkheden en beperkingen van dit programma te kunnen achterhalen.

Damman Software heeft mij hierbij begeleidt en theoretische en praktische aspecten toegelicht van Java. Met deze kennis heb ik een aantal door Damman Software bedachte functies uitgewerkt.

4.3 Installatie methode en instellingen vastleggen

Doordat elk programma via een eigen methode geïnstalleerd moet worden en ingesteld kan worden was het wenselijk dat dit gedocumenteerd werd. Hierdoor kost een herinstallatie minder tijd dan wanneer alles weer opnieuw uitgezocht moet worden.

De documentatie is op de centrale server van Damman Software opgeslagen, zodat het vrij toegankelijk is.

4.4 Usability onderzoek vergelijkbare websites

Door de (nog steeds groeiende) populariteit van het internet en de toename van mogelijkheden om het internet te gebruiken, komen er per dag vele internetsite's bij. De usability van deze pagina's loopt sterk uit een. De ene site is snel en makkelijk in gebruik en bij de ander is het een raadsel hoe je een advertentie kan plaatsen. Om de concurrentie aan te gaan met deze websites moet de advertentiewebsite die ontwikkelt gaat worden in eerste instantie de positieve punten overnemen van de concurrentie en de negatieve punten oplossen.

Aan de hand van dit onderzoek is geprobeerd om achter de positieve en negatieve kanten te komen van de drie bestbezochte advertentiesites van Nederland. De volgende drie websites zijn onderzocht:

- marktplaats.nl
- tweedehands.nl
- ricardo.nl

Allereerst zijn de usability aspecten geselecteerd waarmee een website van alle kanten belicht kan worden. Usability is de mate waarin een interactief systeem de gebruiker in staat stelt effectief, efficiënt en comfortabel in een gegeven omgeving zijn taak kan voltooien.

Na het selecteren van deze aspecten moeten de operationalisaties worden gekozen waarmee getest kan worden. Veel van deze operationalisaties werden al in de bronnen vermeld en zijn ook letterlijk overgenomen.

Voor het testen van de drie websites zijn er gekozen voor de volgende usability-aspecten:

- Robustness

Dit aspect is onder andere te verdelen in twee deelaspecten, namelijk: task conformance en responsiveness.

Task conformance heeft alles te maken met het mentale model van de gebruiker. Wat denkt de gebruiker dat er gebeurt als hij ergens op klikt en wat doet de website als er op wordt geklikt. Komen deze gebeurtenissen overeen?

Responsiveness heeft betrekking op de reactie van de website op een actie van een gebruiker. Als een gebruiker ergens op klikt wordt er dan om een bevestiging van deze actie gevraagd?

- Affect

Bij het aspect affect wordt er gekeken naar de wat de gebruiker van de website vindt. Wordt de gebruiker gefrustreerd door de site of krijgt hij juist voldoening van zijn bezoek.

- Helpfulness

Het usability aspect helpfulness geeft de mate aan waarin de internetsite helpt, en informatie biedt aan de gebruiker. Hierbij gaat het om de vraag of de gebruiker of het duidelijk is wat hij moet doen.

- Effectiveness

De term effectiveness heeft te maken met de effectiviteit van de site. Als een site als doel heeft informatie te verstrekken over weervoorspellingen en de site staat vol met reclames over tv-programma's is de site niet effectief.

- Learnability

Learnability is onder te verdelen in onder andere predictability, synthesizeability en consistency.

Predictability heeft met de voorspelbaarheid van de internetsite te maken. Synthesizeability houdt in dat de gebruiker zowel de gemaakte acties kan verklaren als de gevolgen ervan. Consistency heeft te maken met de consistentie van de website, dat wil zeggen dat de opmaak, dus ook de vormgeving, consistent is doorgevoerd.

- Flexibility

Dit aspect spreekt grotendeels voor zich. Er gekeken naar de flexibiliteit van de internetsite. Het aspect is op te delen in onder andere dialogue initiative en substitutivity.

Dialogue initiative heeft te maken met het feit of de gebruiker de mogelijkheid heeft om te kiezen wat waar gebeurt. Substitutivity wil zeggen dat de gebruikers weten wat zij moeten invullen bij onder andere invoervelden en in welk formaat dat moet gebeuren.

Om het ontstaan van de operationalisaties te verduidelijken wordt het aspect Affect toegelicht.

Operationalisaties van het aspect Affect:

- Heeft de gebruiker een goede eerste indruk van de website?

Bron: <http://www.waller.co.uk/usabilitydetail.htm> , Waller heeft een checklist opgesteld waaronder ook de eerste indruk van de gebruiker.

Waarom toepasbaar: de eerste indruk van een gebruiker is meestal doorslaggevend voor het verdere verloop van zijn bezoek aan de website. Als de gebruiker meteen een goed overzicht heeft is hij tevreden en zal de website verder doorlopen.

- Heeft de website voldoende contrast tussen voor en achtergrond kleur?

Bron: http://www.handboekusability.nl/pdf/usability_checklists_pagina.pdf
De checklist pagina van het handboek usability omschrijft voor de opmaak van de pagina om voldoende contrast te geven aan voor en achtergrond kleur.

Waarom toepasbaar: Om de gebruiker voldoening te geven dient de website helder te zijn, de verdeling van lay-out en tekst dient duidelijk te worden onderscheiden in kleur voor de leesbaarheid.

- Is het duidelijk van wie de website is en wat de website inhoud?

Bron: <http://www.w3use.nl/usability/goudenregels.html> regel een van de gouden regels van w3use. Herkenbaarheid en bedrijfsinformatie.

Bron: De gebruiker heeft een doel voor ogen en wil meteen de juiste informatie hebben over het bedrijf of de instelling. De website dient ook herkenbaar te zijn.

- Is het makkelijk om de pagina's van de website uit te printen?

Bron: <http://www.waller.co.uk/usabilitydetail.htm> regel 2.6 van Waller. Hij zegt hier bijvoorbeeld over: gebruik geen lichte tekstkleuren op een pagina.

Waarom toepasbaar: Vooral van nieuwssites willen mensen vaak een interessant

berichtje printen om te bewaren of om het bericht te delen met anderen.

- Is de linker bovenkant van de pagina gebruikt om navigatie en identiteit weer te geven?

Bron: http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/602 IBM heeft ook een aantal usability eisen opgesteld voor "Visual Layout and elements".

Waarom toepasbaar: IBM heeft deze regel opgesteld vooral uit belang voor de ontwikkeling van software denken wij. Ook voor websites is deze regel toepasbaar omdat mensen graag in een oogopslag zien waar de site over gaat en waar ze heen kunnen. Mensen beginnen met lezen aan de linkerkant.

- Zijn de pagina titels helder en eenduidig?

Bron: <http://www.bunnyfoot.com/freestuff/articles/usability/usabilitychecklist.html>, usability checklist van bunnyfoot.

Waarom toepasbaar: Als de gebruiker de pagina in zijn favorieten zet, komt de pagina titel in de favorieten te staan, vandaar dat dit erg belangrijk is.

- Is de website vrij van pop-up schermen?

Bron:
<http://www.bunnyfoot.com/freestuff/articles/usability/usabilityguidelinesdont.html>, de do's en don't van usability guidelines.

Waarom toepasbaar: De gebruiker heeft graag zelf controle over zijn systeem en wil daarom niet dat er ineens andere schermen openen. De gebruiker kan hier gefrustreerd van raken en zal de website verlaten.

- Worden er teveel animatie gebruikt in de website?

Bron:
<http://www.bunnyfoot.com/freestuff/articles/usability/usabilityguidelinesdont.html>, de do's en don't van usability guidelines.

Waarom toepasbaar: Te veel animaties kan de gebruiker irriteren en kan zorgen voor een lange laadtijd van de website, waardoor de gebruiker snel zal afhaken.

Met deze operationalisaties is een checklist samengesteld waarmee een testpersoon de operationalisaties kan beantwoorden met ja of nee. Als er uitzonderlijke situaties zijn of bijzonderheden kan de testpersoon een motivatie bij elke vraag geven.

De checklisten zijn ingevuld door iemand die geen belang of enige relatie heeft met dit project. Hierdoor wordt een zo objectief mogelijk resultaat bereikt.

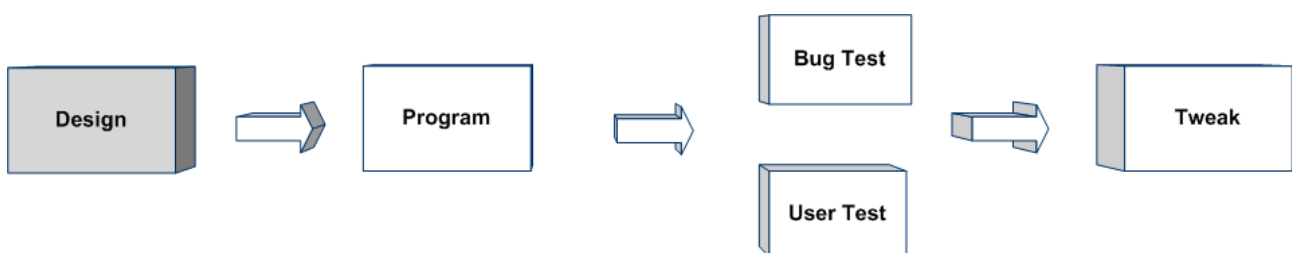
Uit de resultaten van de test (zie Bijlage B: Resultaten Usability onderzoek vergelijkbare websites) valt op dat de drie populaire advertentiewebsites maar weinig negatieve punten hebben. Het grootste nadeel van Marktplaats.nl is het werken met frames. Dit heeft tot gevolg dat de aangeklikte links niet zichtbaar worden in de adresbalk en dat het bij het printen niet vooraf zekerheid is over de content die uitgeprint gaat worden.

Bij het aspect flexibility hebben geen van de drie websites een probleem, of negatieve punten. Voor de rest kan er opgemerkt worden dat tweedehands.nl een erg onduidelijke en rommelige website is. Hierdoor wordt het navigeren door de site moeilijk.

De resultaten van deze test worden gebruikt voor het opstellen van systeemeisen voor de advertentiesite damman.info. Het opstellen van de systeemeisen is een onderdeel van de definitiestudie (zie hoofdstuk 5.2.3 Eisen en wensen opstellen van de gebruikers)

De analysefase is de eerste fase van het ontwikkelproces. De analysefase is een fase waarin keuzes worden gemaakt over de manier waarop de opdracht uitgewerkt zal worden. Deze fase heeft als voordeel dat problemen in een vroeg stadium opgelost kunnen worden voordat er geprogrammeerd wordt. Hierdoor wordt het risico verkleind dat de opdracht niet binnen een bepaalde tijd afgerond kan worden. Tevens worden ideeën van de opdrachtgever en ontwikkelaar op elkaar afgestemd.

Zoals in dit hoofdstuk duidelijk zal worden is er gekozen voor de ontwikkelmethode van Alan Cooper. Cooper heeft het ontwikkelproces in een aantal fases opgesplitst. De eerste fase van Alan, de designfase komt overeen met de analysefase en de conceptfase die gebruikt worden in dit document.



Afbeelding 1 Fase design Alan Cooper

5.1 Opstellen plan van aanpak

Het plan van aanpak is een document waarin de gemaakte keuzes worden vastgelegd wat betreft de aanpak en werkwijze die gebruikt zal worden tijdens de ontwikkeling van de website. Doormiddel van de globale planning (zie Bijlage C: Globale planning) krijgen zowel de opdrachtgever als de ontwikkelaar inzicht in de verwachte tijd die het kost om de website te ontwikkelen. Het plan van aanpak was in principe al uitgewerkt voordat de afstudeeropdracht van start was gegaan. Dit plan was nodig om goedkeuring te krijgen vanuit de Haagse Hogeschool om te mogen starten. Na een aantal bijeenkomsten, vergaderingen en brainstorm momenten zijn we tot de invulling van het plan van aanpak gekomen.

5.1.1 Keuze van methode voor de opdracht

Om een website te ontwikkelen is het van belang dat er een aantal fases worden doorlopen voordat er geprogrammeerd kan worden. Doormiddel van het doorlopen van deze fases moet het voor de ontwikkelaar en opdrachtgever duidelijk worden wat er precies wordt uitgewerkt, op welke manier en hoeveel tijd het gaat kosten.

Een methode, ook wel ontwikkelmethode genoemd, heeft een bepaalde aanpak die gebruikt kan worden voor het uitwerken van een project op een gestructureerde manier.

Tijdens eerdere projecten voor de opleiding VIA is de methode IAD (Het evolutionair ontwikkelen van informatiesystemen) van R.J.H. Tolido gebruikt. De methode IAD die we hebben gebruikt tijdens de opleiding, is ontwikkeld in 1998.

De methode IAD legt naar mijn mening heel erg de nadruk op het iteratief ontwikkelen van een product. Dit houdt in dat bepaalde stappen meerdere keren worden doorlopen om tot een goed resultaat te komen. Ook het uitwerken van een project binnen de gestelde tijd is een belangrijk aspect in IAD. In deze methode wordt stap voor stap aangegeven wat er uitgewerkt moet worden om uiteindelijk het product te kunnen maken. Deze werkwijze is naar mijn mening erg theoretisch opgezet en wordt nauwelijks onderbouwd met praktijkvoorbeelden.

Bij het gebruik van IAD bij voorgaande projecten ben ik ook meerdere malen tegen onduidelijkheden en ,naar mijn mening, nutteloze stappen aangelopen.

Aangezien er bij een afstudeerproject geen eisen werden gesteld aan de methode die gebruikt moest worden ben ik naar een andere methode gaan zoeken. Via een aantal docenten van de opleiding VIA en een zoektocht op het internet ben ik aan de methode van Alan Cooper gekomen.

Alan Cooper is een software-ontwikkelaar die zowel programmeert, ontwerpt en programma's bedenkt. Alan heeft in 1993 zijn eigen bedrijf opgericht in het ontwikkelen van software. Inmiddels zijn er verschillende producten opgeleverd voor bedrijven als IBM, Logitech, Sun Microsystems en 3M.

Het succes van Alan Cooper heeft hij te danken aan zijn gebruikersvriendelijke programma's en doelgerichte aanpak. Om andere softwareontwikkelaars te helpen en natuurlijk om nog meer inkomsten binnen te krijgen, heeft Alan zijn ontwikkelmethode uitgebracht in de boeken "About Face 2.0" en "The Inmates are running the asylum". In deze boeken wordt duidelijk dat Alan veel waarde hecht aan het opstellen van de doelen van een opdracht, het nut van een programma en de eisen en wensen van de gebruikers van het programma. Immers als de doelgroep het programma niet wil gebruiken dan is het ontwikkelen ervan een kostbare aangelegenheid.

De primaire kwaliteiten die Cooper stelt aan de succes van software wordt verduidelijkt in Afbeelding 2 primaire kwaliteiten product volgens Cooper



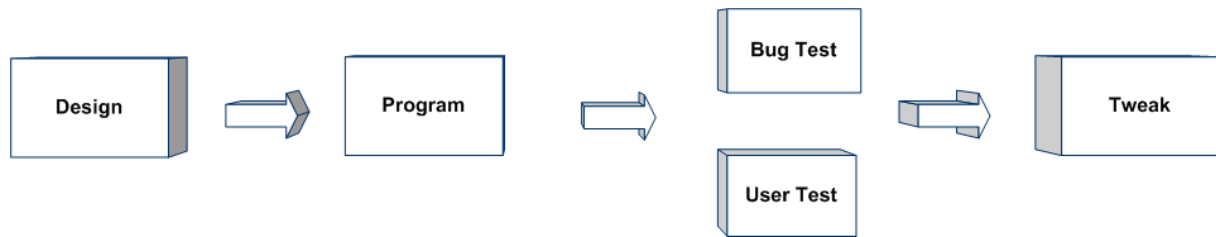
Afbeelding 2 primaire kwaliteiten product volgens Cooper

De stappen van Alan Cooper om tot een goed programma te komen worden verduidelijkt met veel praktijkvoorbeelden. Dit heeft als voordeel dat de doelen van een bepaalde stap snel duidelijk zijn. De methode van Cooper heeft als enige nadeel dat deze geschreven is voor het ontwikkelen van softwareprogramma's en niet zozeer gericht is op de ontwikkeling van een website. Desondanks was ik van mening dat deze methode ook geschikt was voor dit project, mede gebaseerd op mijn vijfjarige ervaring met het ontwikkelen van websites. Een website is immers ook maar een programma wat tegelijkertijd gebruikt kan worden door meerdere gebruikers.

Na het doorlezen van Cooper's boeken heb ik besloten om voor zijn ontwikkelmethode te kiezen. Ik vind het erg belangrijk om een nuttig product af te leveren wat makkelijk in gebruik is en waarbij zowel de gebruiker als opdrachtgever tevreden over kunnen zijn.

5.1.2 Onderverdelen van het ontwikkelproces in fases

De keuze van de ontwikkelmethode heeft als gevolg dat het ontwikkelproces in een aantal fases worden opgesplitst. Aangezien ik gekozen heb voor de methode van Alan Cooper (zie hoofdstuk 5.1.1) wordt het proces opgesplitst in de fases Design, Program, Bug test, User test en Tweak. (zie Afbeelding 3 Fases Alan Cooper) Hieronder volgt een korte uitleg over deze fases.



Afbeelding 3 Fases Alan Cooper

Design

In de designfase wordt het project op papier uitgewerkt. Dit houdt in dat er nagedacht wordt over de doelen van de website, de eisen van de gebruikers en opdrachtgever en de omgeving waar de website wordt gehost. Het doel van deze fase is het nadenken over de problemen, wensen en technische mogelijkheden om de problemen op te lossen. Ook het modelleren van de technische structuur behoort tot de designfase.

Program

De programfase is gericht op het uitwerken van de modellen die gemaakt zijn in de designfase. Het programmeren zal aan de hand van de eisen en wensen worden uitgevoerd zoals deze staan opgesteld in de designfase.

Bugtest

Bij het programmeren is de kans op programmeerfouten aanwezig. Een gevolg hiervan kan zijn de website niet veilig genoeg en dat deze hierdoor makkelijk aangevallen kan worden. Het kan ook voorkomen dat de gegevens die getoond worden niet juist zijn. In de bugtestfase wordt er op deze fouten gecontroleerd.

User test

In de user testfase wordt de website getest door de doelgroep van de website. Dit gebeurt voordat de website in gebruik wordt genomen om eventuele aanpassingen te kunnen doen. Een eerste indruk van een bezoeker is namelijk cruciaal voor het slagen van een website.

Tweak

Het doel van de tweakfase is het aanpassen van de fouten, of veranderingen die zijn opgesteld aan de hand van de user- en bugtests.

5.1.3 Doelen opstellen van de opdracht

In het plan van aanpak is het van belang om erachter te komen wat het doel is van het ontwikkelen van de advertentiewebsite. Damman Software is in dit geval de opdrachtgever en tevens de opdrachtgever voor het ontwikkelen van de website. Het doel van de opdrachtgever is in deze situatie ook het doel van de opdracht.

Zoals al reeds staat omschreven in de opdrachtoomschrijving (zie hoofdstuk 3) is het doel van de opdracht het genereren van inkomsten voor Damman Software.

5.1.4 Opstellen van een globale planning

Omdat deze opdracht binnen de grenzen van een afstudeerproject van VIA afgerond moet zijn is het van belang om in eerste instantie een globale planning te maken. Door deze planning is het mogelijk in de beginsituatie een beeld te krijgen van de globale doorlooptijd van de verschillende fases.

Het inplannen van de verschillende fases is door ervaring met vorige projecten en in overleg met de keurend docent tot stand gekomen.

Deze planning is voornamelijk gebruikt voor het krijgen van een goedkeuring van de Haagse Hogeschool en inzicht te geven in de duur van de verschillende fases voor Damman Software.

5.2 Opstellen definitiestudie

De definitiestudie is een document wat het resultaat is van de analysefase. In dit document staan de resultaten van de gekozen doelgroep, systeemeisen, functionele omschrijving en een pilotontwikkeplan. Dit document dient als basis voor het uitwerken van modellen in het volgende hoofdstuk en het bepalen van de volgorde en tijd voor het programmeren van de website.

5.2.1 Persona's opstellen

Persona's zijn denkbeeldige gebruikers die als hulpmiddel kunnen dienen. die de auteur en software-ontwerper Alan Cooper heeft ontwikkeld. Om een programma zo goed mogelijk aan te laten sluiten bij de gebruikers is een uitgebreide omschrijving van de gebruikers met al hun wensen en eisen het mooist om te hebben. Om dit voor elkaar te krijgen kan je dit het beste aan de gebruikers zelf vragen. Alan Cooper vindt echter dat deze manier van werken door een aantal redenen niet zal werken.

De voornaamste reden die Alan Cooper noemt is dat gebruikers meestal niet in staat zijn voor een probleemstelling een optimale oplossing te bedenken.

De gebruikers zijn wel waardevol, maar voor de ontwikkeling is het volgens de methode van Cooper is het beter om een programma te ontwikkelen voor denkbeeldige gebruikers. Hierdoor worden de oplossingen voor bepaalde problemen niet verstoord door aangedragen oplossingen van echte gebruikers.

De gebruikersgroepen van de advertentiesite vallen in een aantal categorieën onder te verdelen. Per categorie wordt één primaire persona gebruikt voor het ontwikkelen van de website. De persona's zijn in de volgende groepen onderverdeeld:

1. Bezoeker: Een bezoeker is iemand die de website bezoekt, omdat hij of zij op zoek is naar een voorwerp of dienst.
2. Bedrijf: Met bedrijf wordt iemand bedoeld die reclame wil maken op de site voor eigen goederen of diensten.
3. Adverteerder: Een adverteerder is iemand die een bepaald voorwerp of dienst aanbiedt op de website.

Bij de groepen zijn de volgende persona's bedacht.

1. Bezoeker

Pauline Jansen

Pauline is moeder van 3 kinderen die bijna allemaal aan het werk zijn. Ze is altijd huismoeder geweest, maar sinds de kinderen het huis uit zijn is ze gaan werken bij een crèche. Doordat een van haar kinderen haar een pc heeft gegeven is ze begonnen met een verkenningstocht op internet. Haar grootste hobby is het wandelen in de bergen. Doordat ze een beginnende gebruiker is van het internet, is alles vrij nieuw maar wel interessant.

2. Bedrijf

Martijn Borsboom

Martijn (45) is directeur van een Elektro-groothandel in Rotterdam. Hij heeft het bedrijf overgenomen van zijn vader en maakt veel uren om in deze moeizame tijd de omzet en winst in stand te houden. Martijn is erg chaotisch omdat hij alles wel wil doen maar eigenlijk nergens aan toe komt. Thuis is Martijn ook nog lange tijd aan het werk. In het weekend daarentegen gaat hij lekker met zijn vrouw en kinderen naar de hockey en het liefst met de boot varen.

3. Adverteerder

Bert Raaymaker

Bert werkt als verkoper in een kledingzaak en woont met zijn vrouw in Utrecht. Hij werkt vijf dagen in de week en in het weekend gaat hij altijd voetballen. Bert is 34 en voetbalt al sinds zijn achtste. Na het voetballen gaat hij altijd nog even gezellig de kroeg in met zijn team. Sinds de tijd dat hij en zijn vrouw een huis hebben gekocht let Bert wat meer op de centen. Bert houdt niet van oude rommel en biedt daarom regelmatig wat te koop aan.

5.2.2 Doelen opstellen aan de hand van persona's

Doelgericht ontwikkelen is voor Alan Cooper een belangrijk punt. Als je namelijk een programma ontwikkelt wat niet aansluit op het doel van de gebruiker zal het programma nooit gebruikt worden. Een gebruiker moet er immers mee "willen" werken.

Bij de advertentiesite is het eigenlijk voordehandliggend wat de doelen zijn. Ook hierbij zijn de doelen opgesteld aan de hand van de gebruikersgroepen, ofwel de verschillende persona's die zijn opgesteld.

Voor de advertentiesite betekend dit dat de volgende doelen van belang zijn voor de gebruikersgroepen:

bezoeker: het doel van een bezoeker is het kopen van een product waar de bezoeker naar op zoek is.

bedrijf: het doel van een bedrijf is het genereren van inkomsten

adverteerder: het doel van de adverteerder is het verkopen van een product wat hij of zij niet meer wil hebben.

5.2.3 Eisen en wensen opstellen van de gebruikers

Eisen en wensen zijn één van de eerste punten waar je rekening mee zou houden als je een programma ontwikkelt. Om aan deze eisen en wensen te komen is het noodzakelijk om je goed te oriënteren op de doelgroep, in dit geval de opgestelde persona's. De ervaringen van anderen en de ideeën die er zijn om een uniek product neer te zetten.

Het opstellen van deze eisen is gedaan aan de hand van de SMART techniek. SMART staat voor:

1. **S**pecifiek
2. **M**eetbaar
3. **A**anvaardbaar
4. **R**ealiseerbaar
5. **T**ijdsgebonden

Mits een eis aan deze formulering voldoet kan deze getest worden in de testfase en makkelijk ingepland worden om uit te werken. Deze techniek is toegepast als hulpmiddel om de eisen zo volledig mogelijk te formuleren.

Om een eis SMART te maken is niet alleen een goede omschrijving nodig van een eis. De eisen die we voor de advertentiewebsite hebben opgesteld zijn eigenlijk op twee manieren tot stand gekomen. Op de eerste plaats hebben we de eisen en initiële ideeën, die bij aanvang van het project aanwezig waren, getest met de

persona's. Op de tweede plaats is met een persona in gedachten gekeken wat zijn of haar specifieke eis of wens zou kunnen zijn voor het gebruik van de advertentiewebsite.

Een voorbeeld van een eis die we hebben gesteld met de persona Martijn Borsboom (bedrijf) in gedachten is:

- Een bedrijf moet reclame contentgevoelig kunnen plaatsen

Met deze eis wordt bedoeld dat een bezoeker de advertentie pas krijgt te zien als de tekst op de website overeenkomt met het onderwerp van de reclame.

Om de eisen te testen op aanvaardbaarheid is via internet onderzocht of de eis wel wenselijk is voor de gebruiker. Voor de bovenstaande eis is onderzocht of bedrijven nog wel geïnteresseerd zijn in het plaatsen van advertenties op een website en in welke vorm. Relevante artikelen over dit standpunt zijn opgenomen in Bijlage E: Onderzoek aanvaardbaarheid systeemeis. Uit deze artikelen blijkt dat contentgevoelige reclame effectief is en er wordt een groei voorspeld in de uitgaven voor online adverteren en interactieve marketing.

Uit onderzoek is ook gebleken dat deze vorm van reclame niet als irritant gevonden wordt, omdat deze aansluit bij de zoekopdracht. Het is natuurlijk wel van belang de reclame te scheiden van de advertenties voor het bewaren van het overzicht. Een goed voorbeeld hiervan is www.google.nl

Om eisen op te stellen voor alle invalshoeken van een product is er gebruik gemaakt van representative qualities. Deze techniek overkoepelt 12 groepen waarin de eisen kunnen vallen. Het doorlopen van deze 12 groepen zorgt ervoor dat eisen niet specifiek op één gebied worden opgesteld, maar vanuit alle hoeken worden bekeken.

De 12 Representative Qualities zijn:

1. Correctness, Reliability, Robustness
2. Performance
3. User Friendliness
4. Verifiability
5. Maintainability
6. Reusability
7. Portability
8. Understandability
9. Interoperability
10. Productivity
11. Timeliness
12. Visibility

Een uitgebreide uitleg van de representative qualities die gelden voor de advertentiewebsite wordt gegeven in Bijlage F: Representative qualities

Een voorbeeld van een eis die onder de representative quality reusability valt is de volgende:

- De website moet gebruik maken van Cascading Style Sheets (CSS) om de stijl van knoppen, context en tabellen te definiëren

Een volledige lijst met eisen is opgenomen in Bijlage D: Systeemeisen

5.2.4 Functionele omschrijving opstellen

Bij het ontwikkelen van een website met een complexe functionaliteit is het belangrijk een functionele omschrijving te maken. Deze complexiteit zal verder duidelijk worden in de volgende fase bij het modelleren. In een functionele omschrijving wordt in detail uitgewerkt welke acties de gebruiker kan doen en welke gevolgen dat heeft voor het systeem, wat de keuzes zijn die door de gebruiker gemaakt kunnen worden en in welke vorm deze worden afgehandeld.

Aan de hand van deze omschrijving wordt het maken van de modellen in de volgende fase vergemakkelijkt. In de praktijk is het uitwerken van de functionele omschrijving tegelijkertijd gebeurd met het uitwerken van het UML klassendiagram (zie Bijlage G: UML Klassendiagram). Doordat een visuele presentatie meer duidelijkheid verschaft over de structuur en onderlinge relaties is de functionele omschrijving samen met het UML diagram uitgewerkt.

Hieronder volgt een voorbeeld van een functionele omschrijving voor het plaatsen van een advertentie.

Voorbeeld 1 Plaatsen van een advertentie

De algemene gegevens van bij het plaatsen van een advertentie worden opgeslagen in de klasse Advertentie. De specifieke inhoudt van de kenmerken die bij de advertentie horen worden opgeslagen in de klasse advContent. Voorwaarden voor het plaatsen van een advertentie is dat de gebruiker ingelogd is. Aan de hand van de betaaloctie / drempel kunnen er afbeeldingen worden toegevoegd. Deze worden opgeslagen in de klasse advAfbeelding.

De volgende gegevens zullen in de klasse Advertentie worden gezet

- De advertentie krijgt een uniek nummer: advId
- De advertentie krijgt een groeperings nummer waarmee versies bijgehouden kunnen worden, dit nummer is bij plaatsing gelijk aan advId: advIdGroepering.
- Er wordt een verwijzing naar het groeps id (relatieIdGroepering) van de relatie (uit de klasse Relatie) opgenomen in relatieIdGroepering.
- De huidige datum van de plaatsing wordt gezet in: advDatumBegin
- Afhankelijk van de gekozen duur van het plaatsen van de advertentie wordt er een advDatumEind bepaald.
- Er wordt een verwijzing opgenomen van de advertentie groep waar de advertentie in wordt geplaatst in: advGroepId

- Afhankelijk van de betaaloptie wordt er een toonGegevens waarde opgenomen.
 - waarde 0 = toon geen contactgegevens
 - waarde 1 = toon alleen telefoonnummer
 - waarde 2 = toon alleen naw gegevens
 - waarde 3 = toon naw gegevens en telefoonnummer
- Als het gaat om een gratis advertentie of de advertentie is betaald via het prepaid systeem dan wordt de advertentie in eerste instantie op non-actief gezet. Het attribuut actiefStatus is dan 0. Mocht het prepaid saldo niet meer toereikend zijn dan wordt actiefStatus op waarde 5 gezet.
- Er wordt een bevestigingscode aangemaakt die wordt geplaatst in: bevestigCode
- De ingevulde titel van de advertentie wordt opgeslagen in: titel
- De ingevulde prijs van de advertentie wordt opgeslagen in: prijs
- De ingevulde omschrijving van de advertentie wordt opgeslagen in: omschrijving
- De type advertentie, aangeboden (waarde 1) of gevraagd (waarde 2) wordt opgeslagen als: type
- Als er geboden mag worden op de advertentie wordt krijgt het attribuut 'bieden' de waarde 1, anders krijgt deze de waarde 0

De specifieke inhoud van de advertentie zal vastgelegd worden in de klasse advContent. De volgende gegevens zullen hier ingezet worden:

- De verwijzing naar het nummer van de advertentie: advId
- Een uniek nummer: advConId.
- De verwijzing naar de groep van het kenmerk (advKenmIdGroepering) wordt opgeslagen als advKenmIdGroepering
- De ingevulde waarde bij het kenmerk. Bij een radio button of checkbox wordt hier de advGroepKenmId ingevuld om de precieze keuze te kunnen achterhalen van de ingevulde waarde: advContent

Eventuele toegevoegde afbeeldingen worden opgeslagen in advAfbeelding Hierin worden de volgende gegevens opgenomen.

- De afbeelding krijgt een uniek nummer: advAfbeeldingId
- Een verwijzing naar het id van de actieve advertentie: advIdGroepering
- De locatie van de afbeelding in het attribuut bron

Opmerking:

Na het invoeren van de gegevens van de adverteerder, zal er ook gecontroleerd worden op het realtieId in de klasse Adverteerder. Is deze nog niet aanwezig dan zal deze ook worden aangemaakt.

5.2.5 Website opsplitsen in pilots

Bij het ontwikkelen van een website hoort natuurlijk ook het programmeren. Door het hergebruiken van broncode kan de volgorde veel uitmaken bij het opzetten van de website.

Hiervoor is het namelijk noodzakelijk om de code eerst uit te werken alvorens te beginnen met het uitwerken van de pagina's, het beeld wat de gebruiker te zien krijgt.

Een pilot is een losstaand onderdeel van het geheel wat ook als zodanig getest en geïmplementeerd zou kunnen worden. In het geval van de website is de keuze gemaakt om de volgende pilots te gebruiken. Het opstellen van deze pilots was niet geheel onlogisch. Na 5 jaar ervaring in het ontwikkelen van websites ben ik nog steeds van mening dat deze opsplitsing vrij voordehandliggend is.

- Pilot 1 Database installeren en opzetten

In deze pilot zal de databaseserver worden geïnstalleerd, een database worden aangemaakt voor de advertentiewebsite en de tabellen worden aangemaakt.

- Pilot 2 Uitwerken van broncode voor functionaliteit van de website

In pilot 2 wordt de code geprogrammeerd die de acties afhandelt en gegevens ophaalt uit de database. Dit gedeelte zorgt voor de werking van de website.

- Pilot 3 Uitwerken van de code voor de pagina's van de website

Bij de laatste pilot worden de pagina's uitgewerkt die de gebruiker te zien krijgt. Op deze pagina's worden de website visueel in beeld gebracht. Dit zijn voornamelijk invulformulieren of advertenties.

Na het opsplitsen van de pilots is er een planning gemaakt, specifiek om de ontwikkelingsfase nauwlettend in de gaten te kunnen houden, in verband met de deadline. Voor een volledig overzicht van deze planning zie Bijlage I: Planning uitwerken van pilots

Aan de hand van de hoeveelheid weken die in eerste instantie in de globale planning waren opgenomen, heb ik deze verdeelt over de drie pilots. De minste tijd had ik ingepland voor pilot 1, het opzetten van de database. Ik had in de oriëntatiefase de databaseserver al geïnstalleerd en getest. Hierdoor had ik een goede indicatie gekregen wat betreft de duur van deze installatie.

De enige inschatting die nog gedaan moest worden was voor het aanmaken van tabellen voor de database.

Het aanmaken van de tabellen zou in principe maximaal maar één á twee dagen mogen duren. Aangezien het modelleren van deze tabellen al gebeurd was in de conceptfase, was het niets anders dan het UML klassendiagram (zie Bijlage G: UML Klassendiagram) overnemen in de database.

De tweede pilot zou het meeste tijd in beslag nemen, omdat er behoorlijk wat regels code geschreven moest worden en de Java kennis nog vrij beperkt was. Desalniettemin was deze pilot wel één van de belangrijkste om goed uit te voeren, omdat het de gehele werking van de website zou beïnvloeden. Bij onnodig veel broncode zou het verwerken van een aanvraag negatief beïnvloeden. De broncode moest volgens de richtlijnen van Damman Software worden geschreven. (zie Bijlage J: Programeerlijnen)

De derde en laatste pilot bestond alleen uit de opmaak van de website. De pagina's die hierbij gemaakt moesten worden zouden de gegevens tonen die in de objecten waren gezet met acties, geprogrammeerd in pilot 2. Dit zou relatief

gezien weinig tijd in beslag mogen nemen en werd dus ingepland voor twee weken.

5.2.6 Verificatie met de opdrachtgever

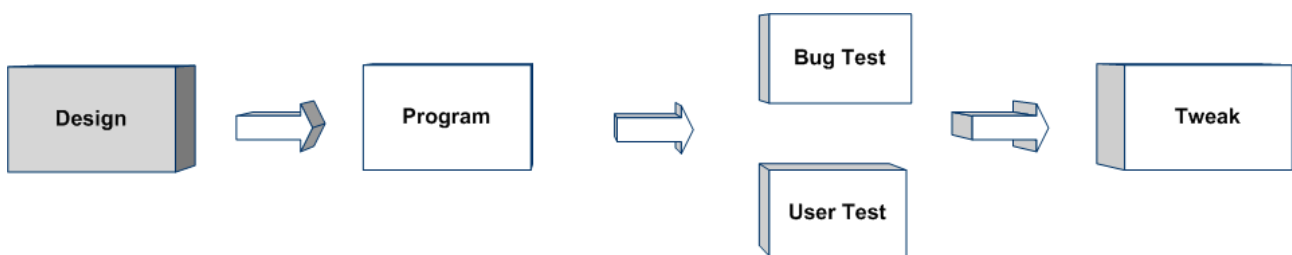
Om tot een goede overeenkomst te komen en afspraken na te komen is het nodig om aan het einde van elke fase de werkzaamheden te verifiëren met de opdrachtgever. Veelal zijn beslissingen in deze fase genomen aan de hand van brainstormen en overleg met Roy Damman, eigenaar van Damman Software en tevens de opdrachtgever voor het ontwikkelen van de advertentiewebsite.

Bij het bespreken van de eindproducten van deze fase, het plan van aanpak en de definitiestudie heb ik allereerst mondeling de veranderingen aangegeven en vervolgens heeft Roy de documenten doorgenomen. De volgende dag werd het commentaar verwerkt en de documenten aangepast om zodoende op één lijn te komen.

De ingrijpendste aanpassing was die van de functionele omschrijving. Deze was naar de mening van Roy niet uitgebreid genoeg om verder te gaan. De functionele omschrijving is belangrijk om de structuur van de broncode goed te onderbouwen. Deze heb ik na een korte bespreking aangepast voordat er begonnen werd aan de volgende fase.

De conceptfase is een fase waarin de technische structuur van de website wordt weergegeven in modellen. Dit wordt gedaan om de relaties te specificeren tussen de verschillende objecten. Tevens vergemakkelijkt dit het opstellen van de onderliggende structuur van de website. Dit hoofdstuk geeft inzicht in de modellen en de gemaakte keuzes hierbij.

Als we kijken naar de fases van Cooper, dan wordt het modelleren gezien als onderdeel van de designfase.



Afbeelding 4 Fase Design van Cooper

6.1 Keuze maken van geschikte modellen

Er zijn verschillende technieken beschikbaar om de technische structuur van een programma te modelleren. Één van deze technieken is UML (Unified Modeling Language). UML is een grafische notatietaal voor het specificeren, visualiseren en documenteren van (complexe) software systemen. UML is onafhankelijk van werkwijze of programmeertaal en is gericht op objectgeoriënteerde systemen.

Andere modelleringstechnieken zijn OMT, Booch en Jacobson. Deze technieken zijn echter verouderd en zijn inmiddels ook door grote bedrijven als IBM en Microsoft vervangen door UML.

UML wordt veel gebruikt in grote projecten. Programmeurs en interface ontwerpers weten aan de hand van de modellen welke relaties er zijn tussen de gebruikersinterface en de onderliggende programmatuur. Letterlijk is UML ook een taal om te kunnen communiceren tussen verschillende ontwikkelaars binnen een project.

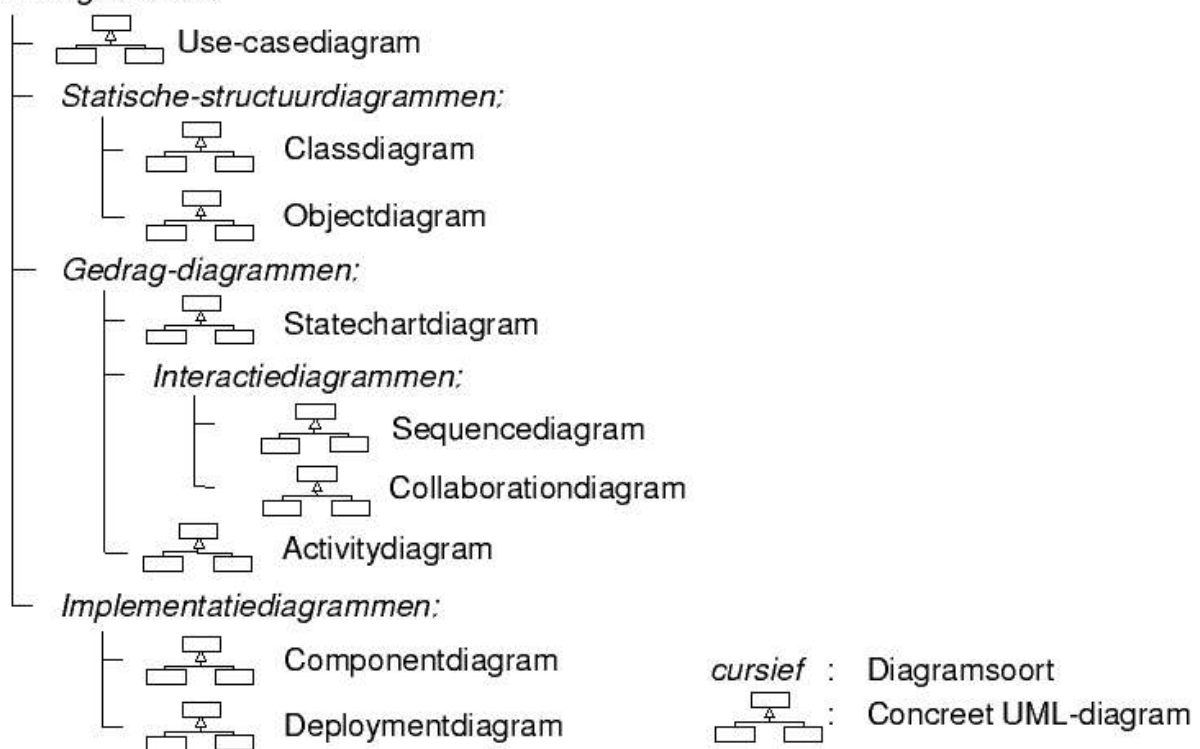
Damman Software vond het wenselijk dat de modellen in UML uitgewerkt zouden worden zodat andere ontwikkelaars zonder moeite aan het project verder zouden kunnen werken.

Zoals te zien is op Afbeelding 5 UML diagrammen zijn er verschillende UML modellen. Bij een project wat zeer groot en complex is, is het wenselijk om al deze modellen uit te werken. Voor het uitwerken van de advertentiesite was deze tijd niet beschikbaar en moesten er keuzes gemaakt worden welke modellen wel en niet zouden worden uitgewerkt.

Aangezien de advertentiesite met veel verschillende gebruikers te maken zou krijgen die elk hun eigen wensen en eisen hebben, was het van belang hier goed over na te denken. Welke scenario's zijn er en hoe het systeem hierop zou reageren. Het use-casediagram was een perfect middel om om dit visueel weer te geven en goed te specificeren.

Een ander belangrijk model is het class diagram, ook wel klassendiagram genoemd. Dit is een model wat gebruikt is om de structuur van de database en benodigde attributen van de klassen te definiëren. De relaties tussen de verschillende klassen worden eveneens in dit model weergegeven.

UML-diagrammen:



Afbeelding 5UML diagrammen

- Use-casediagram: Diagram wat de interactie tussen gebruiker en systeem weergeeft.
- Classdiagram: Beschrijft de statische structuur van een domein of systeem in termen van klassen, hun eigenschappen en onderlinge relaties.
- Objectdiagram: Illustreert een classdiagram door (voorbeeld) objecten en hun eigenschappen en relaties te tonen.
- Statechartdiagram: Model die toestanden, overgangen en gedrag van een object weergeven
- Sequencediagram: Diagram wat de interactie tussen objecten, met de nadruk op de berichten en volgorde van berichten.
- Collaborationdiagram: Diagram die de interactie laat zien tussen objecten. Hierbij ligt de nadruk op (statische) relaties tussen de communicerende objecten.

- Activitydiagram: Geeft de activiteiten weer die nodig zijn voor de voor het uitvoeren van een taak, bijvoorbeeld een use-case.
- Componentdiagram: Een componentdiagram bevat componenten, interfaces en relaties.
- Deploymentdiagram: Het deploymentdiagram toont de fysieke architectuur van een computersysteem. Het kan de computers en randapparatuur beschrijven en de verbindingen ertussen laten zien.

6.2 Modelleren van Use-casediagram

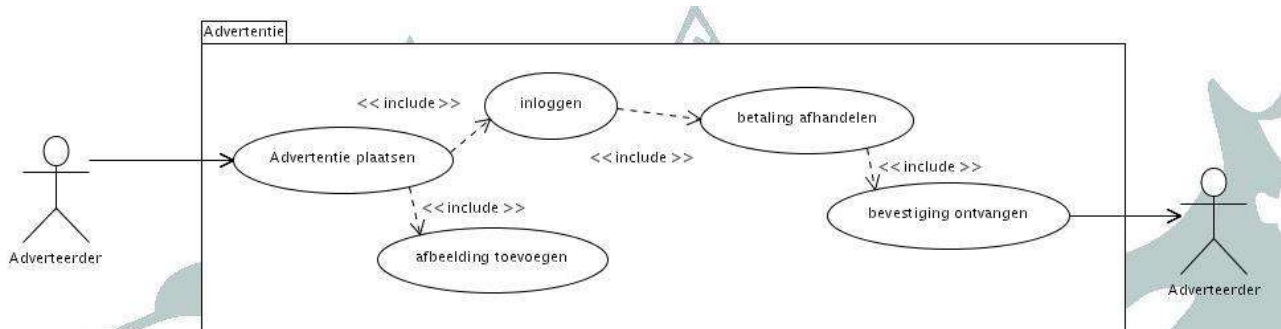
Een use-casediagram laat zien wat er gebeurt als een gebruiker, in dit geval de bezoeker van de website een actie uitvoert. Deze scenario's worden weergegeven doormiddel van een use-casediagram. Voor de advertentiesite zijn alleen de cruciale scenario's uitgewerkt omdat deze van invloed zijn op de werking en structuur van het gehele systeem. Bij het modelleren van een scenario wordt er een omschrijving gegeven en beschreven welke aannames, precondities, postcondities en stappen er zijn.

Als voorbeeld staat hieronder de use-case voor het plaatsen van een advertentie (zie Afbeelding 6 Use-case model Advertentie plaatsen). Een use-case begint bij het nadenken over de stappen die een gebruiker moet doen om een advertentie te kunnen plaatsen, welke resultaten hij krijgt te zien en in welke volgorde. Belangrijk was hierin de keuze betreffende het inloggen.

Om een advertentie te plaatsen dient een bezoeker te zijn ingelogd om te voorkomen dat een bezoeker elke keer zijn naam en andere gegevens moet invullen. Daarnaast kan er achterhaald worden wie de advertentie heeft geplaatst wanneer het gaat om een advertentie die niet voldoet aan de voorwaarden van de site. Ook wordt zo voorkomen dat er op deze manier geen dubbele adverteerders in de database komen te staan.

Het registreren kan een drempel zijn om een advertentie te plaatsen op de website. Hierdoor is in overleg met Damman Software gekozen om het inloggen, mits dit nog niet is gedaan, te laten doen nadat de gegevens van de advertentie zijn ingevuld. Aangezien de bezoeker moeite heeft gedaan om de advertentiegegevens in te vullen is deze eerder geneigd zich te registreren. Er wordt natuurlijk gestreefd naar het maximaal aantal bezoekers en adverteerders om zoveel mogelijk inkomsten te genereren.

Use-case: Een advertentie Plaatsen



Afbeelding 6 Use-case model Advertentie plaatsen

Beschrijving

Het plaatsen van een advertentie op de website

Aannamen

- Er is een database waar de gegevens op geslagen kunnen worden
- De beheerder heeft de instellingen juist geconfigureerd

Precondities

geen

Postcondities

- De adverteerder krijgt een bevestigingsemail gestuurd om de plaatsing te bevestigen

Stappen

1. De adverteerder opent op de website de pagina met het invulformulier om een advertentie te zetten.
2. Er wordt gekeken of de klant is ingelogd. Als de adverteerder niet is ingelogd zal deze alsnog moeten inloggen of zich registreren.
3. Aan de hand van de gegevens en keuzes van de adverteerder wordt er een prijs berekend.
4. Na goedkeuring van de klant wordt de advertentie opgeslagen in de database.
5. Als er een bedrag betaald moet worden, wordt er in eerste instantie gekeken of de adverteerder genoeg saldo heeft. Is dit niet het geval dan wordt de betaling in de database geplaatst met een kenmerk dat de betaling in de wacht staat.
6. Als er voldoende saldo is zal er een email voor bevestiging worden verstuurd naar de adverteerder zodat de advertentie actief wordt.

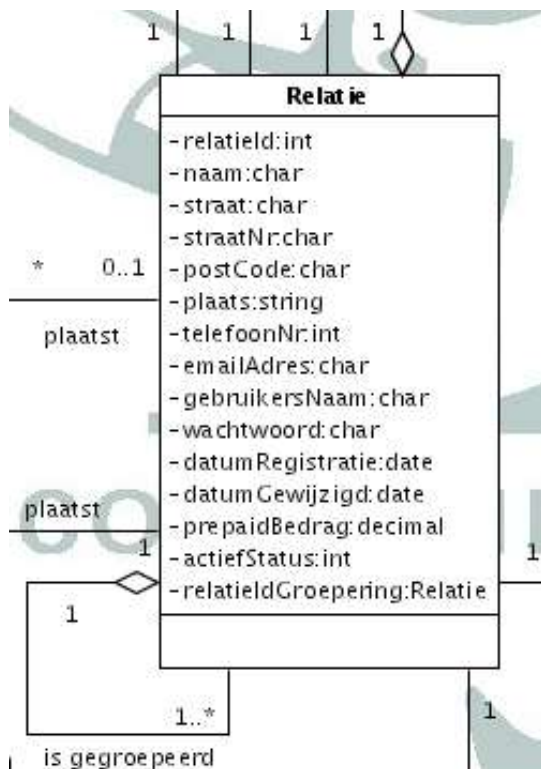
6.3 Modelleren van Classdiagram

Een classdiagram, ook wel een klassendiagram genoemd laat de structuur zien van het systeem met zijn onderlinge relaties. Samen met het uitwerken van de functionele omschrijving (zie hoofdstuk 5.2.5 Functionele omschrijving opstellen) is het classdiagram vormgegeven. In eerste instantie worden de klassen benoemd, waarna de attributen worden toegevoegd. Na het invullen van deze attributen wordt gekeken welke relaties de klassen met elkaar hebben. Hierdoor kan het zijn dat een attribuut van de ene klasse wordt overgenomen in de andere klasse. Zie Bijlage G: UML Klassendiagram voor een uitgewerkte versie van het classdiagram. In overleg met Damman Software is besloten een beperkt gedeelte van dit diagram vrij te geven. Het volledige classdiagram gepresenteerd namelijk de gehele database en het is niet wenselijk dat deze wordt gekopieerd.

Om de werkzaamheden te verduidelijken wordt er als voorbeeld de klasse relatie toegelicht (zie Afbeelding 7 Class Relatie). Om de attributen te bepalen is er gekeken naar de gegevens die in ieder geval nodig zijn voor de werking van de website. Het gaat hierbij om een unieke id (relatieid) en de postcode om de onderlinge afstand tussen adverteerder en bezoeker te kunnen bepalen. Om een relatie te valideren is het noodzakelijk dat deze een emailadres heeft. Daarnaast heeft de relatie een wachtwoord nodig om in te kunnen loggen. Eventueel kan deze ook zijn gebruikersnaam gebruiken. Door een status (actiefstatus) en een versie (relatieidgroepering) toe te voegen kunnen de wijzigingen doormiddel van versies worden bijgehouden voor deze klasse.

Verdere gegevens die de advertentiesite eventueel wil tonen van een relatie zijn de adresgegevens, telefoonnummer, naam en woonplaats. Hiermee kan een bezoeker eventueel contact opnemen met de adverteerder.

Na het invullen van deze attributen en de klassen, worden de relaties bepaald tussen de verschillende klassen. Hierbij wordt gekeken naar het soort relatie en hoe deze aan elkaar gekoppeld kunnen worden. Meestal betekent dit dat het unieke id van de ene klasse wordt overgenomen in de andere klasse. De klasse relatie heeft relaties met de klassen adverteerder, beheerder, bedrijf en prepaidstorting. Voor bijvoorbeeld de klasse prepaidstorting betekent het dat er meerdere stortingen gedaan kunnen worden door één relatie. Dit wordt een één op veel relatie genoemd. Het unieke id van de klasse relatie is daarom opgenomen in de klasse prepaidstorting.



Afbeelding 7 Class Relatie

6.4 Modellen verifiëren met Damman Software

Na het modelleren van de UML diagrammen heb ik de modellen voorgelegd aan Roy Damman. Roy is zelf ook software-ontwikkelaar en heeft samen met mij onder andere een oplossing gezocht voor het bijhouden van wijzigingen van relaties. Dit hebben we opgelost door een attribuut relatieidgroepering aan te maken.

Ook met het oog op de snelheid van het zoeken naar advertenties hebben we gekozen om hiervoor een aparte klasse aan te maken. In deze zoekklasse komen de zoekwoorden te staan waarmee een advertentie geassocieerd kan worden.

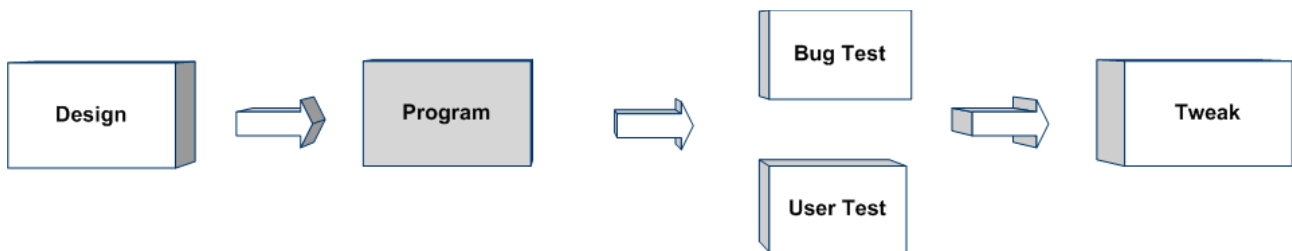
Na het bespreken van de modellen en het noteren van de wijzigingen heb ik de wijzigingen in de modellen doorgevoerd.

In de ontwikkelfase worden de modellen uit de designfase uitgewerkt. Per opgestelde pilot (zie hoofdstuk 5.2.6 Website opsplitsen in pilots) worden de oplossingen uitgewerkt die gevonden zijn voor de opgestelde eisen.

In de tweede en derde pilot waarin de gehele website wordt geprogrammeerd en vormgegeven, worden de oplossingen in volgorde van prioriteit uitgewerkt (zie Bijlage D: Systeemeisen). Dit houdt in dat eerst de functionaliteit wordt uitgewerkt waaraan de prioriteit basis is gegeven. Met deze functionaliteit is er een basisstructuur gemaakt waarmee de website al een alternatief kan zijn voor andere advertentiesites.

Mocht er in de planning van de pilots ruimte over zijn dan wordt de comfort en luxe functionaliteit uitgewerkt.

Als we kijken naar de methode van Cooper dan komt deze fase na de design fase.



7.1 Uitwerken Pilot 1 Database installeren en opzetten

De eerste pilot betreft het installeren van de database voor de website. De structuur van de database is in de conceptfase in het classdiagram al uitgewerkt (zie hoofdstuk 6.3 Modelleren van Classdiagram)

De database heeft eigenlijk maar één doel. Het opslaan van gegevens van de website op een snelle manier en op een centraal punt.

Om een database op te zetten moest er op de eerste plaats een databaseserver geïnstalleerd zijn. Vervolgens moest er een connectie worden gemaakt met deze databaseserver om een database aan te maken en konden de tabellen en attributen worden aangemaakt.

Firebird is als database gebruikt. Aangezien Firebird in de oriëntatiefase al een keer was geïnstalleerd, getest en gedocumenteerd, was het in dit geval niet problematisch om dit nog eens uit te voeren. Dit keer was er, in overleg met Roy Damman, gekozen voor de nieuwste stabiele versie, zodat er geen onverwachte problemen of tekortkomingen zouden optreden.

Het invullen van de database met de nodige tabellen en attributen is gedaan in het programma NetBeans IDE, waarmee de overige twee pilots ook uitgewerkt zijn. Dit programma biedt een duidelijke visuele weergave van de database en je kunt hiermee snel en eenvoudig tabellen aanmaken.

Om de database te testen is er een testpagina opgezet. Met een formulier op een webpagina kunnen gegevens in de database worden ingevoerd en opgehaald. Deze test is gelukt, wat betekent dat de database goed is geïnstalleerd en dat deze gebruikt kan worden voor het verder uitwerken van de andere twee pilots.

7.2 Uitwerken Pilot 2 Uitwerken van broncode voor functionaliteit van de website

In deze pilot zal de basis geprogrammeerd worden voor de website. Na het uitwerken van de basis moet het mogelijk zijn om op een gestructureerde manier functionaliteiten toe te voegen aan de website.

7.2.1 Opzetten en programmeren van de structuur van de broncode

De structuur van de website dient overzichtelijk te zijn. Door een goede structuur moet een actie snel afgehandeld kunnen worden en daardoor sneller een resultaat tonen aan de bezoeker.

Voor het opzetten van deze structuur moesten er een aantal technische problemen worden opgelost. Deze technische problemen kunnen in de volgende categorieën worden ingedeeld.

- Snelheid
- Uitbreidbaarheid en aanpasbaarheid
- Veiligheid

Samen met Roy Damman ben ik gaan brainstormen over de oplossingen van deze vragen.

Optimalisatie van de snelheid

Een eis die we hebben opgesteld is dat de server niet langer dan anderhalve seconde mag gebruiken om een request af te handelen. Winst in snelheid kan op meerdere manieren worden bereikt. Vanaf het moment dat de bezoeker op een button klikt tot en met de tijd dat hij een nieuwe pagina te zien krijgt wordt er een actie afgehandeld. Deze actie houdt in dat er gegevens worden gecontroleerd, eventueel wordt opgevraagd uit de database, een nieuwe pagina wordt gegenereerd en deze wordt verzonden naar de bezoeker.

Deze pilot richt zich met name op de snelheid van het afhandelen van de acties. Een belangrijke keuze die we hebben gemaakt op dit gebied is het gebruik van een connectionpool voor de database. Normaal gesproken wordt er voor elke actie een database connectie gemaakt, en na de actie weer afgesloten.

Aangezien de server bij elke connectie geheugen moet vrijmaken en de connectiegegevens moet controleren is dit onnodig tijdverlies. Een ideale oplossing is het gebruik van een connectionpool. Deze connectionpool zorgt voor aanzienlijke tijdwinst door maar één connectie te maken en deze voor meerdere acties beschikbaar te stellen.

Ook het aantal gelijktijdige bezoekers hebben we beperkt. Dit wordt gelijk bij het eerste bezoek aan de site gecontroleerd. Op deze manier kunnen we een goede snelheid garanderen voor de bezoekers die op dat moment de website bezoeken.

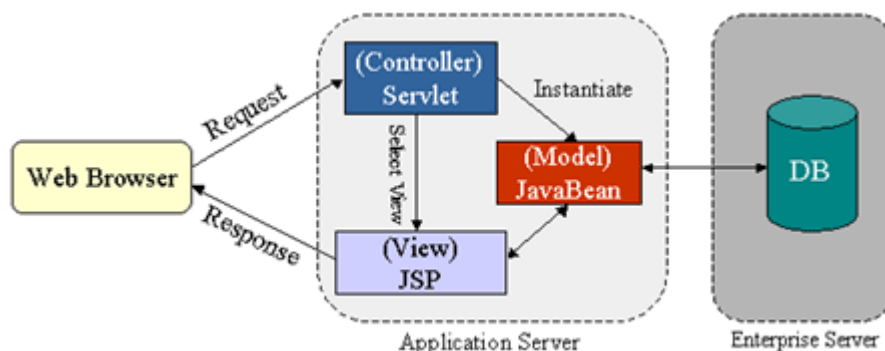
Uitbreidbaarheid en aanpasbaarheid verhogen

Voor het makkelijk uitbreiden en aanpassen van broncode is een goede structuur nodig. De keuze van deze structuur ligt eigenlijk vrij voordehand. Een gescheiden structuur van data, code en weergave.

Deze structuur heeft namelijk als voordeel dat bijvoorbeeld een webdesigner de weergave van de website makkelijk kan aanpassen zonder dat deze verstand heeft van de broncode, of in dit geval de programmeertaal Java. De webdesigner hoeft ook niet te weten hoe de database eruit ziet en waar de gegevens vandaan komen.

Een ander voordeel is het hergebruik van broncode. Het opvragen van een pagina door een bezoeker komt op een centrale plek binnen, waardoor broncode voor het afvangen van fouten en het controleren en toekennen van rechten op één centrale plek gebeurt.

Via het internet ben ik erachter gekomen dat deze structuur veel wordt gebruikt voor professionele projecten en dat deze bekend staat onder de naam Model View Controller (zie Afbeelding 8 Gegevensstroom Model View Controller).



Afbeelding 8 Gegevensstroom Model View Controller

Zoals te zien is op Afbeelding 8 Gegevensstroom Model View Controller komt een aanvraag (request) voor een pagina door een bezoeker binnen bij de controllerservlet. Dit is eigenlijk het centrale punt waar de acties worden afgehandeld.

De controllerservlet controleert in eerste instantie de juistheid van het request, ofwel de url die is verstuurd. Daarna zorgt deze voor een connectie met de connectionpool en wordt het model, mits deze nog niet aanwezig is aangemaakt voor de bezoeker. Vervolgens wordt de actie afgehandeld zoals bijvoorbeeld het opslaan van een advertentie of het opvragen van een adverteerder. Als laatste wordt de pagina opgeroepen die terug wordt gezonden naar de gebruiker. Het model representeert de objecten die gebruikt kunnen worden om gegevens bij te houden uit de database. De jsp pagina's die worden gegenereerd kunnen hier de gegevens uithalen om deze te presenteren aan de bezoeker.

De view is het gedeelte wat de bezoeker te zien krijgt. Een veelgebruikte term hiervoor in de informatica is Grafical User Interface, ook wel afgekort als GUI. De view is eigenlijk de code die gegevens opvraagt uit het model en omzet in htmlcode.

Verhogen van de veiligheid

Voor het verhogen van de veiligheid van de website hebben we er voor gekozen om de gegevens alleen in het model te plaatsen door de controllerservlet. Dit houdt in dat een bezoeker niet rechtstreeks een pagina kan aanroepen, maar deze toegewezen krijgt na het uitvoeren van een actie. Dit houdt in dat er bij alle acties eerst gecontroleerd wordt wie de gebruiker is en of de actie wel uitgevoerd mag worden. Zou de bezoeker dit omzeilen dan krijgt deze gewoon een lege pagina te zien, omdat er geen gegevens in het model zijn geplaatst.

7.2.2 Programmeren van de functionaliteit van de website

Nu er een basisstructuur is gelegd voor de website kan deze worden uitgebreid met acties. De volgorde van het uitwerken van deze acties is afhankelijk van prioriteiten die zijn gesteld. (zie Bijlage D: Systeemeisen)

De acties die gedaan kunnen worden op de website is makkelijk gezegd de functionaliteit van de website. Het zoeken naar een advertentie is bijvoorbeeld zo'n actie.

Voor het uitwerken van deze functionaliteit is behoorlijk wat kennis nodig van Java. Parameters die met het request worden verstuurd moet omgezet worden in het juiste formaat, vergelijkingen worden uitgevoerd en gegevens worden opgehaald uit de database en geplaatst worden in het model.

Om dit te verduidelijken wordt hieronder de actie toegelicht die wordt uitgevoerd als een bezoeker het registratieformulier op de website heeft ingevuld en op opslaan heeft geklikt. In dikgedrukte letters staat de toelichting voor de code eronder.

Voorbeeld 2 Registratie actie

```
// aangeven in welke package de actie zich bevind
package controller;

// bibliotheken importeren
import java.io.*;
import java.sql.SQLException;
import javax.servlet.*;
import javax.servlet.http.*;
import model.*;
import java.sql.*;
import java.util.Date;
import java.util.*;

public class RegistratieSubmitAction extends Action
{
    // Voert de actie uit.

    public void run() throws ServletException, IOException
    {
        // opvragen van de ingevulde waarden van het jspformulier
        String sNaam = request.getParameter("naam");
        String sStraat = request.getParameter("straat");
        String sStraatNr = request.getParameter("straatnr");
        String sPostcode = request.getParameter("postcode");
        String sPlaats = request.getParameter("plaats");
        String sTelefoonNr = request.getParameter("telefoonnr");
        String sEmailAdres = request.getParameter("emailadres");
        String sGebruikersNaam = request.getParameter("gebruikersnaam");
        String sWachtwoord = request.getParameter("wachtwoord");
        String sVoorwaardenOk = request.getParameter("voorwaardenok");

        // variabele initieren voor de actie
        Map _mapFormError = sessie.getErrorMap();
        String sNextUrl = "";

        // gegevens checken en eventueel een medling aanmaken
        if (sNaam == null || sNaam.equals(""))
            _mapFormError.put("naam", "u heeft geen naam ingevuld");
        if (sStraat == null || sStraat.equals(""))
            _mapFormError.put("straat", "u heeft geen straat ingevuld");
        if (sStraatNr == null || sStraatNr.equals(""))
            _mapFormError.put("straatnr", "u heeft geen straatnummer ingevuld");
        if (sPostcode == null || sPostcode.equals(""))
            _mapFormError.put("postcode", "u heeft geen postcode ingevuld");
        if (sPostcode != null && !sPostcode.equals("") && sessie.CheckPostcode(sPostcode) == false)
            _mapFormError.put("postcode", "de ingevulde postcode is niet juist");
        if (sPlaats == null || sPlaats.equals(""))
            _mapFormError.put("plaats", "u heeft geen plaats ingevuld");
        if (sEmailAdres == null || sEmailAdres.equals(""))
            _mapFormError.put("emailadres", "u heeft geen emailadres ingevuld");
        if (sWachtwoord == null || sWachtwoord.equals(""))
            _mapFormError.put("wachtwoord", "u heeft geen wachtwoord ingevuld");
    }
}
```

```

if (sWachtwoord != null && !sWachtwoord.equals("") && sWachtwoord.length() <
5)
    _mapFormError.put("wachtwoord", "het wachtwoord dient langer te zijn dan 4
tekens");
if (sVoorwaardenOk == null)
    _mapFormError.put("voorwaardenok", "u dient accoord te gaan met de
voorwaarden");

// als er geen fouten zijn opgetreden
if (_mapFormError.size() < 1)
{
    // variabele initieren voor de relatie
    Date dDatumRegistratie = new Date();
    float fPrepaidBedrag = 00;
    sPostcode = sPostcode.replaceAll(" ", "");
    int iRelatieIdGroepering = sessie.getRelatieIdGroepering();
    String sAanmeldNaam = "";

    if(sGebruikersNaam == null || sGebruikersNaam.equals(""))
    {
        sAanmeldNaam = sEmailAdres;
    }
    else
    {
        sAanmeldNaam = sGebruikersNaam;
    }

    // aangezien deze actie ook gebruikt wordt voor het wijzigen van
relatiegegevens, eerst controleren wie er is ingelogd
    if(iRelatieIdGroepering < 1)
    {
        // controleren op bestaande gebruikersnaam
        try
        {
            iRelatieIdGroepering = BusinessLogic.getInstance().relatieCheck
(sAanmeldNaam, sWachtwoord);
        }
        catch(SQLException e)
        {
            String sServletError = e.toString();
            _mapFormError.put("ServletError", sServletError);
        }

        if(iRelatieIdGroepering > 0 && sGebruikersNaam != null && !
sGebruikersNaam.equals(""))
            _mapFormError.put("gebruikersnaam", "gebruikersnaam is al in gebruik,
probeer een andere naam");

        if(iRelatieIdGroepering > 0 && sEmailAdres != null && !sEmailAdres.equals
(""))
            _mapFormError.put("emailadres", "emailadres is al in gebruik, probeer
een andere naam");
    }
    if (_mapFormError.size() < 1)
    {
        if(sTelefoonNr == null || sTelefoonNr.equals(""))
            sTelefoonNr = "0";

        if(sGebruikersNaam == null)
            sGebruikersNaam = "";
    }
}

```

```

        try
        {
            // nieuwe relatie aanmaken.
            BusinessLogic.getInstance().newRelatie(sNaam, sStraat, sPostcode,
            sPlaats, sTelefoonNr, sEmailAdres, sGebruikersNaam, sWachtwoord,
            sStraatNr, dDatumRegistratie, fPrepaidBedrag, iRelatieIdGroepering);
        }
        catch(SQLException e)
        {
            String sServletError = e.toString();
            _mapFormError.put("ServletError", sServletError);
        }
    }
}
// als er een fout is opgetreden
if (_mapFormError.size() > 0)
{
    // terug sturen naar pagina waar de request vandaan is gekomen.
    sNextUrl = "/registratie.jsp";
}
else
{
    // Als alles klopt wordt de volgende pagina aangeroepen
    sNextUrl = "/resultaat.jsp";
}
RequestDispatcher requestDisp = application.getRequestDispatcher(sNextUrl);
if (requestDisp == null)
    throw new ServletException("Kan" + sNextUrl + "niet vinden");

requestDisp.forward(request, response);
}
}

```

7.3 Uitwerken Pilot 3: Uitwerken van de code voor de pagina's van de website

In deze pilot worden de pagina's uitgewerkt die zorgen voor de weergave van de website. Als we kijken naar Afbeelding 8 Gegevensstroom Model View Controller dan wordt de view genoemd. In de vorige pilot is de basisstructuur en de functionaliteit geprogrammeerd. De view haalt de gegevens uit het model en zet deze om in html met behulp van de scripttaal jsp.

Ook voor deze pilot is snelheid een belangrijk aspect om rekening mee te houden. Na het afhandelen van een actie wordt de htmlpagina gegenereerd en verzonden naar de bezoeker. Om een pagina zo snel mogelijk terug te sturen is het belangrijk dat de pagina zelf zo klein mogelijk wordt gehouden. Hiermee wordt bedoeld dat plaatjes niet te groot zijn en van op een juiste manier gecomprimeerd.

In overleg met Roy Damman is de keuze gemaakt om de website op te delen in drie frames. Hierdoor haalt de bezoeker alleen bij de eerste keer de lijst met advertentiegroepen op. Door deze opsplitsing wordt er ongeveer 50kb minder data

verzonden naar de bezoeker. Met een lichte adslverbinding scheelt dit al bijna 2 seconde voor elke actie die uitgevoerd wordt.

Omdat bij deze pilot ook veel gebruik gemaakt kan worden van hergebruikte code is een goede structuur ook belangrijk. Elke pagina is dan ook opgesplitst in een aantal gedeeltes:

Het eerste gedeelte is de header. Hierin wordt informatie zoals metatags, de titel van de site en de link naar een stylesheet opgehaald.

Het tweede gedeelte is het ophalen van het menu. Aan de hand van de rechten van de gebruiker zullen de links en opties voor bepaalde pagina's wijzigen.

Het derde gedeelte van de pagina is het contentgedeelte. Dit kan een resultaat zijn van een zoekactie, een formulier om gegevens in te voeren of een bevestiging voordat iets in de database wordt geplaatst. De content kan globaal worden onderverdeeld in de volgende punten. In het contentgedeelte worden de gegevens uit het model opgevraagd en in htmlcode omgezet.

Het vierde gedeelte van de pagina is de footer, hiermee wordt een standaard footer opgehaald met bijvoorbeeld een copyright.

Deze opsplitsing heeft als voordeel dat alleen het veranderende gedeelte van een pagina, de content geprogrammeerd hoeft te worden. De header, links en footer hoeven hierdoor niet elke keer gekopieerd te worden. Dit komt de aanpasbaarheid ten goede.

Om dit te verduidelijken wordt hieronder de pagina toegelicht die gepresenteerd wordt als een bezoeker is ingelogd. In dikgedrukte letters staat de toelichting voor de code eronder.

Voorbeeld 3 JSP pagina

```
<%-- verschillende bibliotheken raadplegen --%>
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core_rt" %>
<%@ taglib prefix="sql" uri="http://java.sun.com/jstl/sql_rt" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@taglib prefix="my" tagdir="/WEB-INF/tags/mytags" %>
<%@ page import="model.*" %>
<%@taglib prefix="my" tagdir="/WEB-INF/tags/mytags" %>

<%-- voeg de header en links toe --%>
<jsp:include page="header.jsp" />
<jsp:include page="hoofdlinks.jsp" />
```

```
<%-- check of er is ingelogd --%>
<my:checklogin />

<%-- als er niet is ingelogd wordt deze terugverwezen naar de inlogpagina --%>
<c:if test="${relatietype == 'geen'}">
<jsp:forward page="/login.jsp">
<jsp:param name="sError" value="u dient eerst in te loggen" />
</jsp:forward>
</c:if>

<%-- het model ophalen van de bezoeker --%>
<jsp:useBean id="sessie"
    scope="session"
    class="model.Sessie"/>

<% request.setAttribute("relatie", sessie.getRelatie()); %>

<jsp:useBean id="relatie"
    scope="request"
    class="model.Relatie" />

<%-- de naam van de bezoeker opzoeken die is ingelogd in het model --%>
Welkom! ${relatie.naam}

<%-- voeg de footer toe --%>
<jsp:include page="footer.jsp" />
```

Na het uitwerken van de pilots in het vorige hoofdstuk moet het systeem worden getest. Hierbij kunnen programmeerfouten worden weggewerkt en beveiligingslekken worden gedicht. In deze fase wordt er op twee manieren getest. In de eerste test wordt er getest op programmeerfouten en fouten in de functionele werking van de website, deze test is voor een groot deel door de ontwikkelaar zelf gedaan.

De tweede test wordt uitgevoerd door echte potentiële gebruikers van de website. Het doel is om de usability te testen en deze eventueel nog aan te scherpen.

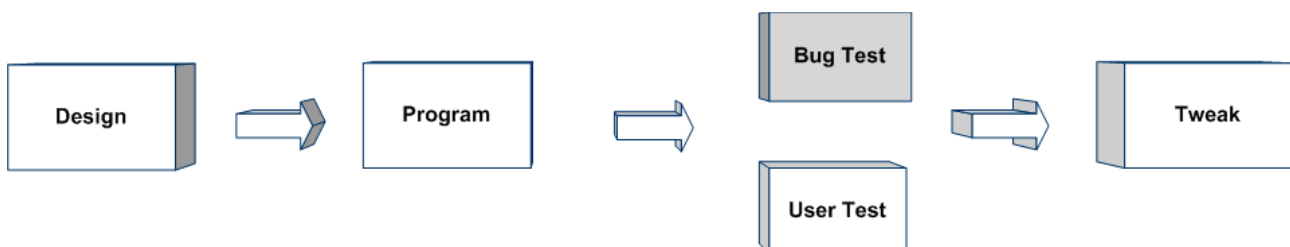
Als resultaat zal uiteindelijk een wijzigingsrapport worden opgesteld waar de wijzigingen in staan die verbeterd moet worden voordat de website officieel geïmplementeerd kan worden.

Aangezien de tweede pilot is uitgelopen op de planning hebben we niet de mogelijkheid gehad om het gehele systeem te kunnen testen. In overleg met Roy Damman hebben we besloten om het plaatsen van een advertentie uitgebreid te testen. Deze functionaliteit is een onderdeel van het vernieuwende gedeelte van de website, omdat hierbij verschillende kenmerken naar voren worden gehaald die specifiek zijn voor een advertentiegroep. De programmatuur die zorgt voor het ophalen van deze kenmerken wordt tevens gebruikt bij het geavanceerd zoeken naar een advertentie. Door test zal dus uiteindelijk een belangrijk gedeelte van het systeem worden getest.

8.1 Testen van de website op bugs

Het testen op bugs houdt in dat de broncode wordt gecontroleerd op fouten. Verwijzingen naar objecten worden nagekeken en de broncode wordt gecontroleerd op juistheid. Het doel van deze test is het verhogen van de betrouwbaarheid en de voorspelbaarheid van de site.

Als we kijken naar de methode van Alan Cooper dan komt de Bug Test na de programfase.



Afbeelding 9 Bug Testfase Cooper

Er zijn twee technieken om software op bugs te testen. Dit zijn Black-Box Testing en White Box Testing.

8.1.1 Testen van website met Black Box Testing

Black Box testing wordt ook wel functioneel testen genoemd. Dit is een testtechniek waarbij de interne werking van het systeem niet bekend is bij de tester. De tester weet bijvoorbeeld alleen de ingevoerde waarde en wat de verwachte uitkomst moet zijn. Hij weet niet hoe de ingevoerde waardes worden verwerkt. De tester heeft dan ook geen enkel belang bij de kennis van de code van het systeem.

De voordelen van deze techniek zijn:

- De test is objectief, omdat de ontwikkelaar en de tester onafhankelijk van elkaar zijn.
- De tester heeft geen kennis nodig van de programmeertaal of specificaties van het systeem
- De test wordt gedaan vanuit het oogpunt van de gebruiker en niet de ontwikkelaar
- De testscenario's kunnen al worden opgesteld als de specificaties van het systeem klaar is

De nadelen van deze techniek zijn:

- De test kan overbodig zijn als de ontwikkelaar al een aantal scenario's heeft getest.
- De testscenario's zijn moeilijk op te stellen
- Het testen van alle mogelijke ingevulde waardes is onmogelijk, want dit zo een enorme hoeveelheid tijd kosten. Hierdoor zullen lang niet alle scenario's getest kunnen worden

Voor het testen van de website met deze techniek is er eerst een testscenario opgesteld. Dit scenario richt zich op het plaatsen van een advertentie, de ingevoerde waardes worden na het klikken op de knop "verder" gecontroleerd door het systeem. En als uitkomst wordt een voorbeeld van de advertentie getoond met dezelfde waardes die zijn ingevoerd. Het gebruikte testscenario wordt hieronder vermeld, met daaronder het invulformulier voor het plaatsen van een advertentie en het voorbeeld.

De uitslag van de test was positief bevonden. De tester had geen op- of aanmerkingen en de uitvoer was zoals verwacht.

Voorbeeld 4 Testscenario advertentie plaatsen

Tescase:	Advertentie plaatsen
Testpersoon:	Onno Schut
Invoer	Uitvoer
Hoofdgroep: Auto's Subgroep: Volvo Titel: zgan S40 2003 Omschrijving: Europa(common rail)bj.3-2000 Auto verkeerd in zeer goede staat. Distr.riem vervangen bij 155000km. Onderhoudspapieren aanwezig.Graag telefonische reacties. Bieden: Nee Type: Aangeboden Prijs: 8500.00 Kleur: Rood Opties: Airco Aandrijving: Handgeschakeld KM Stand: 179000	Auto >> Volvo Aangeboden: zgan S40 2003 Omschrijving: Europa(common rail)bj.3-2000 Auto verkeerd in zeer goede staat. Distr.riem vervangen bij 155000km. Onderhoudspapieren aanwezig.Graag telefonische reacties. Kleur: Rood Opties: Airco Aandrijving: Handgeschakeld KM Stand: 179000 Geplaatst op: 09-06-2004 15:17:40 Prijs: € 8500.00

Plaats hier uw advertentie

Advertentie plaatsen

Hoofdgroep

Auto

Subgroep

Volvo

Titel *

zgan S40 2003

Omschrijving

goede staat.
 Distr.riem vervangen bij
 155000km.
 Onderhoudspapieren
 aanwezig.Graag
 telefonische reacties.

Bieden

Nee

Type

Aangeboden

Prijs

8500,00 Euro

Kleur

Rood

Opties *

☒ Airco
☐ Licht metalen velgen

Aandrijving *

☐ Automaat
☒ Handgeschakeld

KM Stand *

179000 KM

* = verplicht

Legen

Verder

Afbeelding 10 Invulformulier advertentie plaatsen

Voorbeeld van de advertentie

Auto >> Volvo

Aangeboden	zgan S40 2003
Omschrijving	Europa(common rail)bj.3-2000 Auto verkeer in zeer goede staat. Distr.riem vervangen bij 155000km. Onderhoudspapieren aanwezig.Graag telefonische reacties.
Kleur	Rood
Opties	Airco
Aandrijving	Handgeschakeld
KM Stand	179000
Geplaatst op	09-06-2004 15:17:40
Prijs	€ 8500.0

Terug

Verder

Afbeelding 11 Voorbeeld advertentie

8.1.2 Testen van website met White Box Testing

White Box Testing, wat ook wel Glass Box, Structural of Open Box Testing wordt genoemd is een testtechniek waarbij volledige kennis van de broncode nodig is van het gedeelte wat getest wordt. Anders dan met black box testing gebruikt white box testing specifieke kennis van de broncode om de uitvoer te analyseren.

De test kan alleen nauwkeurig worden uitgevoerd als de tester precies weet hoe het programma werkt. Op deze manier kan hij zien of de systeem wel op de juiste manier reageert en waar dan eventueel de fout zich bevindt.

In het ontwikkelprogramma NetBeans IDE zit een perfect gereedschap om de white box test uit te voeren. Hiermee kan je op elke wenselijke plek van de broncode de gegevens bekijken van de variabele die op dat moment aanwezig zijn. Hiermee kan je exact nagaan of het systeem de juiste gegevens doorvoert en waar de eventuele fouten zich bevinden.

Hieronder is een voorbeeld te zien van de white box test die gedaan is met het scenario wat opgesteld is bij de black box test (zie Testscenario advertentie plaatsen, pagina 48)

Local Variables		Watches	Call Stack
Name		Value	
_mapFormError		#1924	...
listAdvContent		#1926	...
listAdvGroepKenmerk		#1921	...
listAdvHoofdGroep		#1923	...
listAdvSubGroep		#1922	...
mapAdvContent		#1927	...
sAdvBiedBedragMinCent		"00"	...
sAdvBiedBedragMinEuro		"00"	...
sAdvBieden		"0"	...
sAdvKenmerk		"1"	...
sAdvOmschrijving		"Europa(common rail)bj.3-2000\r\nAuto verk...	...
sAdvPlaatsSubmit		"Verder"	...
sAdvPrijsCent		"00"	...
sAdvPrijsEuro		"8500"	...
sAdvTitel		"zgan S40 2003"	...
sHoofdgroepId		"1"	...
sNextUrl		""	...
sSubgroepId		"2"	...
this		#1910	...

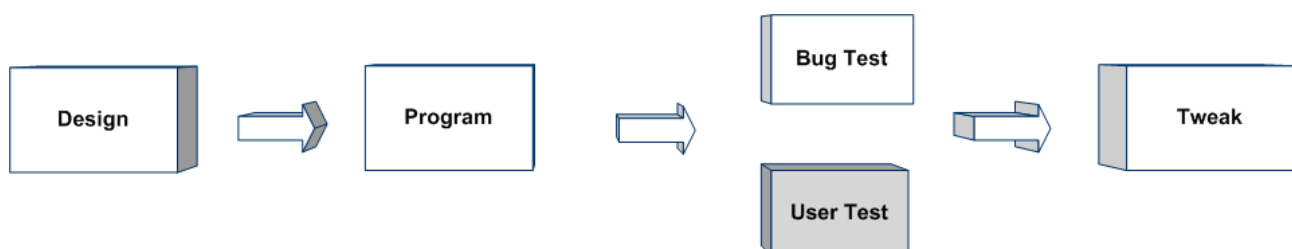
Afbeelding 12 White Box Testing met Netbeans IDE

8.2 Testen van de website aan de hand van persona's

In deze paragraaf zal de website worden getest aan de hand van de opgestelde persona's uit de definitiefase (zie **hoofdstuk ..**) Het doel van deze test is het achterhalen van mogelijke probleemgebieden voor één van de doelgroepen die de persona's moeten vertegenwoordigen.

Net als bij de voorgaande tests zal ook deze gericht zijn op het plaatsen van een advertentie. Het verschil met de vorige tests is dat het hier om de usability gaat. Weet de gebruiker wel wat hij of zij moet doen?

Als we kijken naar de methode van Cooper dan wordt deze test uitgevoerd ongeveer op dezelfde tijd als de bug test die hiervoor is gedaan.



Afbeelding 13 User Testfase Cooper

Voor het testen met een persona heb ik een objectief persoon, die niets met dit project te maken heeft, de persona Bert Raaymaker in gedachten laten nemen. Bert Raaymaker is de enige persoon die op dit moment deze test uit kan voeren omdat hij de adverteerders vertegenwoordigt.

De testpersoon heeft de instructie gekregen om een advertentie te plaatsen op de site met de gegevens die ook zijn ingevoerd bij de bug test. Hierbij ging het niet zozeer om het goede resultaat, maar meer om de plaatsing van de knoppen en invoervelden. Ook moet het duidelijk zijn wat de persona moet invullen.

Aangezien Bert niet van rommel houdt werd er vooral gekeken naar de lay-out van de pagina. Na het selecteren van een subgroep werden de velden zichtbaar die bij deze groep horen. Aangezien de opmaak en lay-out van de pagina niet veranderde leidde dit niet tot ergernis.

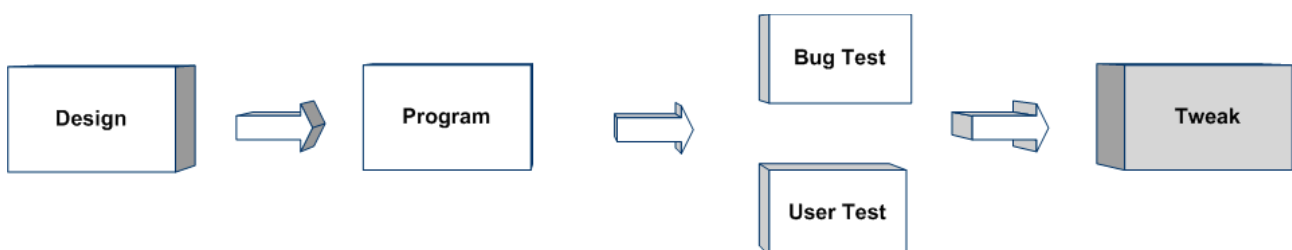
Ook was vrij duidelijk wat hij moest invullen bij de verschillende invoervelden. De kenmerken die verplicht waren werden duidelijk aangegeven.

Op deze test met Bert Raaymaker als adverteerder zijn geen op of aanmerkingen geplaatst. Helaas is de website nog niet in het stadium dat deze in zijn geheel getest kan worden. Deze test zegt dan ook alleen nog maar iets over een beperkt onderdeel van de website.

8.3 Opstellen van een wijzigingsrapport

In het wijzigingsrapport staan de oplossingen die gedefinieerd zijn aan de hand van de fouten en problemen die bij de tests naar voren zijn gekomen. Hiermee wordt een overzicht gegeven van de aanpassingen die gedaan moeten worden. Aangezien er uit zowel de User Test en de Bug Test geen fouten of moeilijkheden zijn gekomen was het niet noodzakelijk om een wijzigingsrapport op te stellen.

Na het opstellen van het wijzigingsrapport worden normaal gesproken de wijzigingen doorgevoerd. Door Alan Cooper wordt dit ook wel de Tweakfase genoemd. Zie onderstaande afbeelding.



Afbeelding 14 Tweakfase Cooper

In dit hoofdstuk wordt de gehele projectperiode geëvalueerd. Er wordt hierbij onderscheidt gemaakt in twee evaluaties:

- Evaluatie van het gevolgde proces
- Evaluatie van de website

Beide evaluaties worden besproken in onderstaande hoofdstukken

9.1 Evaluatie van het gevolgde proces

Het proces is de manier waarop een product tot stand is gekomen. In dit hoofdstuk geef ik een mening over het gevolgde proces. Hierbij zullen ook de problemen of positieve punten naar voren komen. In de paragraaf 9.1.1 wordt de gebruikte methode van Alen Cooper geëvalueerd. In paragraaf 9.1.2 worden de gebruikte technieken besproken.

9.1.1 Evaluatie van de gebruikte methode

Alan Cooper heeft mij vanaf het begin van de afstudeerperiode al een goede manier van denken aangeleerd. Vanuit de opleiding was het aangeleerd om zoveel mogelijk volgens planning te werken en de producten op tijd af te leveren. Alan Cooper heeft door zijn methodiek veel meer de nadruk gelegd op de doelen van de gebruikers, financiële belangen van de organisatie en de technische mogelijkheden om het product uit te werken. Door deze instelling ben ik samen met Damman Software tot een doelgericht product gekomen.

Een fase waar normaal gesproken niet zoveel tijd aan besteed wordt is de oriëntatiefase. Juist door deze fase wist ik de mogelijkheden van Java en kon ik vroegtijdig de programmeertaal verkennen. In die fase heb ik juist een basis gekregen die voor de rest van het project erg belangrijk is geweest.

De analysefase is een fase uitstekende fase geweest waarin de systeemeisen zijn opgezet en de functionele structuur. Allebei deze aspecten zijn belangrijk geweest voor het programmeren en testen van de website.

In de conceptfase zijn de use-case en de class diagrammen gedefinieerd. Deze UML diagrammen waren bedoeld om inzicht te geven in de relaties van de achterliggende structuur van de website. Aangezien de structuur nogal complex was door de unieke eigenschappen van de website kwam er veel duidelijkheid bij het opstellen van deze diagrammen. Zo konden ook de gedachten van de Roy Damman en mij goed op elkaar worden afgesteld.

De pilotontwikkelfase is een fase geweest waarin de website eigenlijk geprogrammeerd is. Door de complexiteit van de structuur en de ervaring heeft deze fase langer geduurd dan in eerste instantie gepland was. Hierdoor is de laatste fase, de testfase in het gedrang gekomen.

De testfase is uiteindelijk maar een klein gedeelte geweest van het project. Dit kwam ondermeer omdat er niet veel te testen was. Het belangrijkste gedeelte is echter wel getest en met deze uitslag kan er tevreden worden teruggekeken op het project. Dat wat geprogrammeerd was is goed gegaan, alleen de ingeplande tijdsduur was uiteindelijk niet haalbaar.

9.1.2 Evaluatie van de gebruikte technieken

De vooraf opgestelde technieken waren niet altijd wenselijk. Het houden van workshops was bij dit project niet noodzakelijk en aangezien er ook geen tijd voor was is dit deze techniek ook niet gebruikt.

Een techniek die echter wel goed heeft gewerkt is UML. De uitgewerkte use-case en class diagrammen hebben het opstellen van een functionele omschrijving en het definiëren van objecten vergemakkelijkt. Hierdoor werd aanzienlijke tijd bespaart bij het programmeren van de website. Aangezien ik weinig ervaring had met zowel jsp als Java werd deze winst teniet gedaan en is de site helaas niet compleet afgerond.

Een andere techniek die erg geholpen heeft bij het proces is de SMART formulering. Deze techniek heeft ervoor gezorgd dat de systeemeisen duidelijk gedefinieerd zijn en realistisch. Door het onderzoeken van de aanvaardbaarheid van een eis werd al in de analysefase duidelijk of het een nuttige eis zou zijn of niet.

Het werken met persona's was nieuw voor mij, maar het is eigenlijk een onmisbare techniek voor het ontwikkelen van software. Echte gebruikers hebben altijd een mening en willen daarbij vaak al een oplossing aandragen voor een probleem waar eigenlijk niet over nagedacht is. Persona's zorgen ervoor dat je de software wel ontwikkelt voor een doelgroep, maar je niet laadt afleiden door een bestaande gebruiker. Hierdoor worden meestal betere oplossingen bedacht.

Normaal gesproken had ik al een denkbeeldige doelgroep en gebruiker in gedachten bij het ontwikkelen van software. Echter werd er deze keer een naam aan gegeven en werd het daadwerkelijk een persoon bedacht.

9.2 Evaluatie van de website

De website is het product wat uiteindelijk de inkomsten moet genereren voor Damman Software. Echter is de website niet geheel afgerond. Dit is deels te wijten aan de ervaring die niet aanwezig was van Java, en anderzijds heb ik wellicht de complexiteit van het systeem onderschat. De ideeën zijn echter tot in detail uitgewerkt en meerdere malen geëvalueerd. Hierdoor is er een goede basis uitgewerkt waarmee Damman Software verder kan.

Na mijn verdediging van dit afstudeerproject zal ik in ieder geval nog een basisversie afronden waarmee Damman Software in ieder geval kan gaan concurreren met de andere advertentiesites.

Advertentie

Een advertentie is een tekstuele omschrijving van een product wat iemand wil verkopen of op zoek naar is.

Adverteerder

Een adverteerder is iemand die een advertentie plaatst, of heeft geplaatst en wordt dan ook als eigenaar van de advertentie gezien.

Advertentie Alarm

Met een advertentie alarm kan een geregistreerde worden gewaarschuwd, bijvoorbeeld via email als er een advertentie is geplaatst. De geregistreerde heeft dan wel 1 of meerdere voorwaarden ingesteld waar deze advertentie aan moet voldoen.

Advertentie groep

Een advertentie groep kan een hoofd of subgroep zijn waar een advertentie bij geplaatst kan worden. Zo kan een advertentie wat gaat over een mountainbike geplaatst worden onder een groep fietsen.

Advertentie kenmerken

Elke advertentie kan in een groep worden ingedeeld, bijvoorbeeld bij de hoofdgroep auto's en bij de subgroep volkswagen. Bij de subgroep volkswagen horen bepaalde kenmerken. Een voorbeeld van een kenmerk is "type" of "aantal deuren". Deze specifieke kenmerken voor een advertentie subgroep worden advertentie kenmerken genoemd.

Geregistreerde

Een geregistreerde is iemand die bij de website geregistreerd staat. Dit kan een adverteerder zijn, een beheerder, een bedrijf of iemand die nog niet is ingedeeld in een groep, maar wel in de klasse Relatie is opgenomen.

Afbeelding

Als afbeelding wordt gezien een plaatje met de extensie .jpg of .gif

Reclame

Reclame is een afbeelding, of tekstuele aankondiging wat geplaatst is door een bedrijf. Het doel van reclame is het vergroten van de naamsbekendheid van het bedrijf of het aankondigen van een actie, aanbieding bij het bedrijf.

Bedrijf

een bedrijf is een commerciële organisatie, onderneming.

Bezoeker

Een bezoeker is iemand die niet is ingelogd op de site. Hierdoor is het niet vast te leggen of het hier gaat om een adverteerder, bedrijf, of beheerder.

Bod

Een bod is een bedrag wat een geregistreerde uitbrengt op een advertentie waar op

geboden mag worden.

Beheerder

Een beheerder is iemand die de website beheerd en onderhoudt. In eerste instantie zal de beheerder de advertentie groepen aanmaken en prijzen en regels vaststellen. Bij de advertentiesite wordt een beheerder onderkent als degene die is ingelogd bij het relatieId van de klasse Beheerder overeenkomt met het relatieId bij zijn inlognaam en wachtwoord.

Prepaid

Prepaid is een account waar geld op gestort kan worden door een geregistreerde. Met deze account kunnen verschillende betaalde acties worden uitgevoerd zonder daarvoor elke keer een bedrag te moeten storten, overmaken of via creditcard te moeten betalen.

Prepaid saldo

Met het prepaid saldo wordt het huidige prepaid bedrag bedoeld wat de gebruiker nog overheeft.

Hits

Een hit is een keer dat een bezoeker of geregistreerde iets heeft gezien. Hits is een aantal keer dat iets is getoond, bijvoorbeeld een afbeelding heeft 23 hits, betekend dan dat de afbeelding 23 keer is getoond.

Reactie

Een reactie kan verstuurd worden via de website om contact op te nemen met de adverteerder, of om meer informatie te weten te komen over de advertentie.

Invoerveld

Een invoerveld is een onderdeel van een formulier waar een bezoeker, adverteerder of beheerder een waarde kan invullen die vervolgens verwerkt kan worden. Hier kan het gaan om een tekst voor het zetten van een advertentie of het kiezen uit verschillende opties.

Invoerveld type

Een invoerveld heeft een bepaald type. Dit kan bijvoorbeeld een checkbox zijn of een textarea. Hiermee dwing je de gebruiker om een formulier op een bepaalde manier in te vullen.

LiteratuurLijst

NIELSEN, J.

Designing Web usability, New Riders Publishing USA, 2000, ISBN 1-56205-810-X.

Phil Hanna

The Complete Reference JSP 2.0, Osborne, 2003, ISBN 0-07-222437-1

Douglas Bell, Mike Parr

Java voor studenten 3e editie, 2002, ISBN 90-430-0515-0

Internet adressen

http://www.d-sciencelab.com/siteview/1_guides/1_tree/index.html

<http://www.benvroom.nl/pretestenchecklistbeknopt.htm>

<http://www.w3use.nl/usability/goudenregels.html>

<http://www.handboekusability.nl/checklists/>

http://www.handboekusability.nl/pdf/usability_checklists_NietDoen.pdf

<http://usability.pagina.nl/>

<http://www.useit.com/>

<http://www.w3.org/TR/WAI-WEBCONTENT/>

<http://www.waller.co.uk/usabilitydetail.htm>

http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/602

<http://www.bunnyfoot.com/freestuff/articles/usability/usabilitychecklist.html>

http://searchwebservices.techtarget.com/originalContent/0,289142,sid26_gci893132,00.html

<http://www.drempelsweg.nl/>

<http://www.drempelsweg.nl/toegankelijkbouwen/richt/richtlijnen.php3?w3c=ja>

<http://www.theserverside.com/articles/article.tss?l=BestBothWorlds>

<http://www.devx.com/webdev/Article/16973/1954?pf=true>

<http://www.ingester.nl/Meting%20R16.htm>

<https://www.google.com/adsense/policies>

<http://www.cutthecrap.biz/software/whitepapers/alchemist.html>

http://www.idevelopment.info/data/Programming/java/PROGRAMMING_Java_Programming.shtml

<http://www.regexlib.com/Search.aspx?k=html>

<http://javaalmanac.com/egs/javax.servlet.jsp/myform.jsp.html?l=new>

<http://scott.yang.id.au/archives/000423.php>

<http://www.informit.com/articles/article.asp?p=30946&seqNum=4>

http://www.webdevelopersjournal.com/columns/connection_pool.html

<http://www.javacommerce.com/faqs/servletfaq.htm>

<http://archives.java.sun.com/cgi-bin/wa?A2=ind9904&L=jsp-interest&F=&S=&P=18741>

<http://archive.coreservlets.com/Chapter18.html>

<http://courses.coreservlets.com/Course-Materials/#CSAJSP2>

http://www.caucho.com/articles/jsp_templates.xtp

http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc_p.html
<http://www.aanbodpagina.nl/>
<http://www.marktnet.net/>
<http://www.geocrawler.com/archives/3/440/2000/1/0/3218940/>
<http://www.treffend.nl/index.php>
<http://www.kraampjes.nl/>
<http://www.rtl.nl/actueel/rtlnieuws/>
<http://www.x-store.nl/marktplaats/>
<http://www.openoffice.org>

Bijlage A: Volledige opdrachtsomschrijving

Marc Schut

Kenmerk: DOA2004-1.1.152

Wijzigingen in de student- en/of bedrijfsgegevens (j/n):

zo ja, welke:

Omschrijving aangepast door:

Ontwikkeling van een advertentiewebsite bij Damman Software

Damman Software is een bedrijf dat is opgericht in 1992. Het bedrijf werkt voor het midden- en kleinbedrijf en verkoopt en ontwikkelt maatwerksoftware. Damman Software is ook behulpzaam bij het uitbouwen en koppelen van bestaande systemen geleverd door derden.

Damman Software's specialiteit ligt ondermeer bij logistieke systemen inclusief koppeling met bestaande systemen.

Momenteel zijn er veel websites gericht op het aanbieden van tweedehands goederen. Deze websites zijn vaak niet consistent qua opzet en bieden niet de gewenste zoekmogelijkheden. Het is de bedoeling een advertentiewebsite te maken die deze Websites overtreft in het gebruiks- en zoekgemak.

Doordat er gebruik gemaakt wordt van kenmerken per advertentiecategorie kan er via een uitgebreide zoekfunctie gericht worden gezocht dan bij de huidige advertentiewebsites. Damman Software denkt dat dit gemak extra bezoekers, adverteerders en bedrijven aan zal trekken. Hierdoor kunnen er tarieven gevraagd worden voor advertenties, reclame en aanvullende diensten.

Daar het beoogde aantal bezoekers groot is, moet getracht worden het dataverkeer zo beperkt mogelijk te houden. Dit moet worden bereikt zonder dat dit ten koste gaat van de gebruikerservaring.

De beheerder zal via een beheerdersgedeelte de site kunnen instellen en de advertenties en gebruikers kunnen beheren. Om ervoor te zorgen dat er geen ongewenste advertenties worden geplaatst moet er een manier worden gevonden om dit tegen te gaan. Op dit moment wordt er gedacht aan een bevestigingssysteem via email.

Het doel van de website is het genereren van inkomsten voor Damman Software. Hiervoor moet de website worden voorbereid op het invoeren van commerciële toepassingen. Er wordt hierbij gedacht aan het betalen voor advertenties, reclame en aanvullende diensten. In eerste instantie ligt de nadruk op het aantrekken van

zoveel mogelijk bezoekers en adverteerders. Hierdoor wordt het in een later stadium makkelijker om bedrijven aan te trekken en inkomsten te genereren.

Het doel van de afstudeeropdracht is het ontwikkelen van een advertentiewebsite. Deze website moet gericht kunnen zoeken op advertenties dan bestaande advertentiewebsites. Om inkomsten te kunnen genereren moet de website worden voorbereid op commerciële toepassingen. Een randvoorwaarde van de site is het beperken van onnodig dataverkeer en de belasting van de server.

Door de ontwikkelomgeving gelijk te stellen aan de productieomgeving van de site zullen de opstartproblemen zo beperkt mogelijk worden gehouden. Dit houdt in dat de Website wordt ontwikkeld op basis van Java Servlets, Java Server Pages draaiend op een Linux, Apache machine. Als Relationeel DataBase Management Systeem wordt gebruik gemaakt van Firebird. De keus voor Firebird is gemaakt aan de hand van goede ervaringen van Damman Software. Firebird is een veelzijdige en gratis database management systeem dat goed presteert met Linux als operating system. Als ontwikkelprogramma wordt gebruik gemaakt van NetBeans. De domeinnaam www.indemarkt.info is reeds geregistreerd en kan gebruikt worden om de site op zetten en te testen.

Bij Damman Software is voldoende kennis in huis om de opdracht goed te kunnen ondersteunen. Er zijn ook meerdere voorbeelden van andere reeds ontwikkelde sites die eventueel gebruikt kunnen worden voor deze opdracht.

De student heeft bij aanvang van de afstudeerperiode een basiskennis Java, geen kennis van Linux en NetBeans. De student heeft wel veel ervaring in het ontwikkelen van Websites met een database koppeling.

De volgende software zal gebruikt worden:

- Linux.
- Apache.
- Netbeans.
- OpenOffice.
- Firebird RDBMS.
- Java SDK 1.4.2.

In het kader van de afstudeeropdracht zullen de volgende activiteiten verricht worden:

Oriëntatiefase:

- Verdiepen in Linux.
- Verdiepen in Java.
- Onderzoeken usability van soortgelijke Websites.
- Opstellen plan van aanpak.
- Verifiëren plan van aanpak met de opdrachtgever.

Analysefase:

- Definiëren doelgroep en de bijbehorende persona's.

- Definieren van de systeemeisen.
- Verifiëren van de opgestelde systeemeisen met de opdrachtgever.
- Bepalen van de structuur van de prototypes.
- Definieren van de style-guide.

Conceptfase:

- Modelleren taakdiagrammen.
- Modelleren UML-klassendiagram.
- Modelleren UML-UseCasediagram.
- Modelleren UML-componentendiagram.
- Uitwerken van een prototype van de Grafical User Interface (GUI).

Ontwikkelingfase:

- Opzetten van de database.
- Programmeren van de broncode, volgens de programmeerstijlrichtlijnen van het bedrijf.
- Verifiëren van de geprogrammeerde onderdelen met opdrachtgever.

Test- en implementatiefase:

- Testen op bugs.
- Testen van de usability aan de hand van persona's.
- Aanbrengen van eventuele verbeteringen aan de hand van de twee bovenstaande tests.
- Samenstellen van de systeemdokumentatie.

De methode van Alan Cooper zal gebruikt gaan worden als basis. Alan Cooper heeft een methode ontwikkeld waardoor het uiteindelijke product meer op de gebruiker wordt toegespitst. Dit heeft hij gedaan door bestaande, veelgebruikte software te analyseren.

Er zullen verschillende technieken van GUIDE gebruikt worden. UML wordt op verzoek van Damman Software gebruikt om de verschillende modellen te ontwikkelen. Een opsomming van de belangrijkste technieken volgt hieronder:

- UML-Klassendiagram.
- UML-UseCasediagram.
- UML-Componentendiagram.
- Taakdiagrammen.
- Time-Boxing, door het in delen opzetten van de systeemeisen, kan er meer controle worden verkregen op de planning en uitvoering hiervan.
- Representative qualities: systeemeisen die verduidelijkt worden door deze in bepaalde categorieën in te delen. Met de categorieën wordt hier onder andere gebruikersvriendelijkheid en prestaties bedoeld.
- Brainstormen.
- SMART (Specifiek, Meetbaar, Aanvaardbaar, Realiseerbaar en Tijdsgebonden). De SMART techniek zorgt ervoor dat systeemeisen op een juiste manier worden opgesteld.
- Workshops.

De nadruk bij de Website ligt bij de werking van de unieke zoekmogelijkheid, voorbereidingen op commerciële toepassingen en het beperken van onnodig dataverkeer. De site moet snel, consistent en overzichtelijk zijn. De documentatie dient volledig en consistent te zijn uitgewerkt.

De volgende producten zullen opgeleverd worden:

- Usability onderzoek.
- Plan van aanpak.
- Definitiestudie.
- Ontwikkelplannen.
- Testrapport.
- Website.
- Systeemdokumentatie.

Bijlage B: Resultaten Usability onderzoek vergelijkbare websites

	Website: http://www.marktplaats.nl/			
		Ja	Ne e	Motivatie
1	Is de pagina opgedeeld in frames?	X		
2	Is de url gecompliceerd?		X	Nee, maar de url is niet zichtbaar in het browser venster.
3	Wordt er voldoende uitleg, hulp gegeven over de navigatie?		X	
4	Zijn de namen van de links informatief. Geven ze informatie over de pagina die wordt geopend na het aanklikken?	X		
5	Heeft de pagina een soortgelijke werking als andere internetpagina's in dezelfde categorie?	X		
6	Is het om de website te kunnen bekijken noodzakelijk om de nieuwste plugins te downloaden?		X	
7	Laden de pagina's snel?	X		
8	Is de vormgeving van de site rustig en overzichtelijk?	X		
9	Heeft de gebruiker een goede eerste indruk van de website?	X		Ja, door de rustige presentatie van de vele gegevens maakt de pagina een goede indruk.
10	Heeft de website voldoende contrast tussen voor en achtergrond kleur?	X		
11	Is het duidelijk van wie de website is en wat de website inhoud?	X		Het is duidelijk wat de website inhoud, maar van wie de website is en wat hun doel is, is totaal onduidelijk.
12	Is het makkelijk om de pagina's van de website uit te printen?		X	Nergens is een print knop te vinden, en er wordt gebruik gemaakt van frames dus is het lastig te bepalen welk frame wordt geprint.
13	Is de linker bovenkant van de pagina gebruikt om navigatie en identiteit weer te geven?	X		
14	Zijn de pagina titels helder en eenduidig?	X		Er wordt duidelijk aangegeven in welke categorie je je bevindt. Alleen de titel in de browser wordt nooit vervangen. Dat is een minpunt.
15	Is de website vrij van pop-up schermen?	X		
16	Worden er teveel animaties gebruikt in de website?		X	
17	Is er een horizontale scrollbar zichtbaar op de website?		X	
18	Is het lettertype goed en duidelijk leesbaar?		X	Voor mensen met minder goede ogen is de tekst zo nu en dan moeilijk te lezen.
19	Als de site meer dan honderd pagina's telt, is er een zoekfunctie geïmplementeerd?	X		
20	Is er een sitemap aanwezig op de website?		X	Is ook niet gewenst op deze site. Vanwege de steeds wisselende advertenties.
21	Kun je vanaf de pagina waar je nu op bevindt je alle pagina's op de site openen?		X	Alleen hoofdmenu items en submenu items van de actieve hoofdmenu item.
22	Is de website toegankelijk gemaakt voor iedereen?	X		
23	Is de pagina ook zonder kleur toonbaar?	X		
24	Geeft de uitkomst van de zoekmachine op de site duidelijk informatie over de gevonden pagina's. En waarom deze zijn gevonden?	X		Er wordt gelijk informatie over de gevonden items getoond. Je kan al gelijk beslissen of dit een relevante uitkomst voor je is of niet.
25	Is de informatie op de website actueel?	X		
26	Bevinden er zich grote lappen teksten zonder highlights, sub koppen en bullets op de pagina?		X	
27	Is de sitemap geplaatst op een rustige pagina, zonder bijkomende onderdelen?		X	Geen sitemap aanwezig.
28	Zijn alle pagina's ingevuld, of staat er ergens "under construction"?		X	Geen underconstruction aanwezig.
29	Is er een afbeelding in de achtergrond van de tekst verwerkt?		X	

30	Is de vormgeving van de site functioneel en ondersteunend in plaats van dominant?	X		De vormgeving is heel rustig en accentueert de nodige onderdelen van de site.
31	Staat er voldoende informatie over de organisatie op de site?		X	
32	Maakt de site geen amateuristische indruk door middel van spel- en taalfouten of andere slordigheden?		X	De advertenties vallen onder verantwoordelijkheid van individuele gebruikers.
33	Sluiten de inhoud en de presentatie van de website aan op de doelen van de organisatie?	X		
34	Is er op elke pagina hetzelfde navigatieschema gebruikt?	X		
35	Worden er zogenaamde "breadcrumbs" gebruikt zodat de gebruiker weet waar hij zich bevindt?	X		
36	Is er een heldere omschrijving van waar de link naartoe wijst?	X		
37	Kan de gebruiker altijd de back/terug button gebruiken?	X		
38	Kan de gebruiker eenvoudig zien in welk formaat hij data in kan voeren?	X		Er is commentaar geplaatst bij de invoervelden als dat nodig is.
39	Is het mogelijk voor gebruikers om tijdsgebonden acties(scripts) te beheersen?		X	Geen scripts aanwezig.
40	Is de site in meerdere typen browsers zichtbaar?	X		
41	Creëer geen periodiek zelfverversende pagina's totdat user agents de mogelijkheid bieden die zelfverversing te stoppen.	X		
42	Is ervoor gezorgd dat alle informatie die met behulp van kleur wordt overgebracht ook beschikbaar is zonder kleur, bijvoorbeeld uit de context of uit de opmaak?	X		Alles is met tekst toegelicht.
43	Worden er shortcuts aangeboden voor belangrijke links?	X		
44	Is de pagina in meerdere talen te lezen?		X	

	Website: http://www.tweedehands.nl/			
		Ja	Ne e	Motivatie
1	Is de pagina opgedeeld in frames?		X	
2	Is de url gecompliceerd?		X	Er is gebruik gemaakt van sub-domeinen. Erg duidelijk.
3	Wordt er voldoende uitleg, hulp gegeven over de navigatie?	X		Wordt wel uitleg gegeven, maar is niet overzichtelijk.
4	Zijn de namen van de links informatief. Geven ze informatie over de pagina die wordt geopend na het aanklikken?	X		
5	Heeft de pagina een soortgelijke werking als andere internetpagina's in dezelfde categorie?	X		
6	Is het om de website te kunnen bekijken noodzakelijk om de nieuwste plugins te downloaden?		X	
7	Laden de pagina's snel?	X		
8	Is de vormgeving van de site rustig en overzichtelijk?		X	
9	Heeft de gebruiker een goede eerste indruk van de website?		X	
10	Heeft de website voldoende contrast tussen voor en achtergrond kleur?	X		
11	Is het duidelijk van wie de website is en wat de website inhoud?	X		Er wordt een aparte pagina getoond met 'eerste keer' waarin uitleg wordt gegeven. Erg duidelijk.

12	Is het makkelijk om de pagina's van de website uit te printen?		X	Er is geen print knop, maar de pagina bestaat wel uit een stuk dus het is voorspelbaar wat er geprint gaat worden.
13	Is de linker bovenkant van de pagina gebruikt om navigatie en identiteit weer te geven?	X		
14	Zijn de pagina titels helder en eenduidig?	X		
15	Is de website vrij van pop-up schermen?	X		
16	Worden er teveel animaties gebruikt in de website?		X	
17	Is er een horizontale scrollbar zichtbaar op de website?		X	
18	Is het lettertype goed en duidelijk leesbaar?	X		
19	Als de site meer dan honderd pagina's telt, is er een zoekfunctie geïmplementeerd?	X		Maar geen handige.
20	Is er een sitemap aanwezig op de website?	X		
21	Kun je vanaf de pagina waar je nu op bevindt je alle pagina's op de site openen?		X	
22	Is de website toegankelijk gemaakt voor iedereen?	X		
23	Is de pagina ook zonder kleur toonbaar?	X		
24	Geeft de uitkomst van de zoekmachine op de site duidelijk informatie over de gevonden pagina's. En waarom deze zijn gevonden?		X	Rommelig overzicht, niet handig.
25	Is de informatie op de website actueel?	X		
26	Bevinden er zich grote lappen teksten zonder highlights, sub koppen en bullets op de pagina?		X	
27	Is de sitemap geplaatst op een rustige pagina, zonder bijkomende onderdelen?	X		
28	Zijn alle pagina's ingevuld, of staat er ergens "under construction"?		X	Geen under construction aanwezig.
29	Is er een afbeelding in de achtergrond van de tekst verwerkt?		X	
30	Is de vormgeving van de site functioneel en ondersteunend in plaats van dominant?		X	Net iets te dominant, rustigere kleuren hadden beter geweest.
31	Staat er voldoende informatie over de organisatie op de site?	X		
32	Maakt de site geen amateuristische indruk door middel van spel- en taalfouten of andere slordigheden?		X	
33	Sluiten de inhoud en de presentatie van de website aan op de doelen van de organisatie?	X		
34	Is er op elke pagina hetzelfde navigatieschema gebruikt?	X		
35	Worden er zogenaamde "breadcrumbs" gebruikt zodat de gebruiker weet waar hij zich bevindt?		X	
36	Is er een heldere omschrijving van waar de link naartoe wijst?	X		
37	Kan de gebruiker altijd de back/terug button gebruiken?	X		
38	Kan de gebruiker eenvoudig zien in welk formaat hij data in kan voeren?		X	
39	Is het mogelijk voor gebruikers om tijdsgebonden acties(scripts) te beheersen?		X	Geen scripts aanwezig.
40	Is de site in meerdere typen browsers zichtbaar?	X		
41	Creëer geen periodiek zelfverversende pagina's totdat user agents de mogelijkheid bieden die zelfverversing te stoppen.	X		
42	Is ervoor gezorgd dat alle informatie die met behulp van kleur wordt overgebracht ook beschikbaar is zonder kleur, bijvoorbeeld uit de context of uit de opmaak?	X		
43	Worden er shortcuts aangeboden voor belangrijke links?	X		

44	Is de pagina in meerdere talen te lezen?		X	
----	--	--	---	--

	Website: http://www.ricardo.nl/			
		Ja	Ne e	Motivatie
1	Is de pagina opgedeeld in frames?		X	
2	Is de url gecompliceerd?		X	Kort en bondig
3	Wordt er voldoende uitleg, hulp gegeven over de navigatie?		X	Er staan categorieën, niet helder dat daar de items onder zitten
4	Zijn de namen van de links informatief. Geven ze informatie over de pagina die wordt geopend na het aanklikken?	X		
5	Heeft de pagina een soortgelijke werking als andere internetpagina's in dezelfde categorie?	X		Ja de veilingen worden aangeboden per categorie zoals ook bij andere marktplaatsen.
6	Is het om de website te kunnen bekijken noodzakelijk om de nieuwste plugins te downloaden?		X	
7	Laden de pagina's snel?	X		
8	Is de vormgeving van de site rustig en overzichtelijk?	X		Ja veel wit ruimte voor de tekst, niet teveel kleur, het tonen van resultaten is alleen wat rommelig
9	Heeft de gebruiker een goede eerste indruk van de website?	X		
10	Heeft de website voldoende contrast tussen voor en achtergrond kleur?	X		
11	Is het duidelijk van wie de website is en wat de website inhoudt?	X		Ja op de homepage staat dat het een online veiling betreft
12	Is het makkelijk om de pagina's van de website uit te printen?		X	Nee er is voor advertenties geen printbutton.
13	Is de linker bovenkant van de pagina gebruikt om navigatie en identiteit weer te geven?	X		
14	Zijn de pagina titels helder en eenduidig?	X		
15	Is de website vrij van pop-up schermen?	X		
16	Worden er teveel animaties gebruikt in de website?		X	
17	Is er een horizontale scrollbar zichtbaar op de website?	X		
18	Is het lettertype goed en duidelijk leesbaar?	X		Beetje klein maar wel duidelijk
19	Als de site meer dan honderd pagina's telt, is er een zoekfunctie geïmplementeerd?	X		
20	Is er een sitemap aanwezig op de website?		X	
21	Kun je vanaf de pagina waar je nu op bevindt je alle pagina's op de site openen?		X	Advertenties staan onder categorieën.
22	Is de website toegankelijk gemaakt voor iedereen?	X		
23	Is de pagina ook zonder kleur toonbaar?		X	
24	Geeft de uitkomst van de zoekmachine op de site duidelijk informatie over de gevonden pagina's. En waarom deze zijn gevonden?	X		
25	Is de informatie op de website actueel?	X		
26	Bevinden er zich grote lappen teksten zonder highlights, sub koppen en bullets op de pagina?	X		
27	Is de sitemap geplaatst op een rustige pagina, zonder bijkomende onderdelen?		X	
28	Zijn alle pagina's ingevuld, of staat er ergens "under construction"?	X		Alle pagina's zijn ingevuld.
29	Is er een afbeelding in de achtergrond van de tekst verwerkt?		X	
30	Is de vormgeving van de site functioneel en ondersteunend in plaats van dominant?	X		Bij het tonen van de resultaten worden er wel teveel verschillende kleuren gebruikt en is de layout niet consistent. Over het algemeen is de vormgeving redelijk functioneel

31	Staat er voldoende informatie over de organisatie op de site?	X		
32	Maakt de site geen amateuristische indruk door middel van spel- en taalfouten of andere slordigheden?		X	Slordig is het verspringen van het scherm wanneer je tussen de hoofditems navigeert
33	Sluiten de inhoud en de presentatie van de website aan op de doelen van de organisatie?	X		
34	Is er op elke pagina hetzelfde navigatieschema gebruikt?		X	
35	Worden er zogenaamde "breadcrumbs" gebruikt zodat de gebruiker weet waar hij zich bevindt?		X	
36	Is er een heldere omschrijving van waar de link naartoe wijst?	X		
37	Kan de gebruiker altijd de back/terug button gebruiken?	X		
38	Kan de gebruiker eenvoudig zien in welk formaat hij data in kan voeren?		X	
39	Is het mogelijk voor gebruikers om tijdsgebonden acties(scripts) te beheersen?		X	
40	Is de site in meerdere typen browsers zichtbaar?	X		
41	Creëer geen periodiek zelfverversende pagina's totdat user agents de mogelijkheid bieden die zelfverversing te stoppen.	X		Niet toegepast op de website
42	Is ervoor gezorgd dat alle informatie die met behulp van kleur wordt overgebracht ook beschikbaar is zonder kleur, bijvoorbeeld uit de context of uit de opmaak?	X		
43	Worden er shortcuts aangeboden voor belangrijke links?	X		Ja op de homepage staan links naar speciale veilingen.
44	Is de pagina in meerdere talen te lezen?		X	Er is wel voor elk land een ricardo site

Bijlage C: Globale planning

Oriëntatiefase: duur 3 weken

- Week 1 (9 / 13 febr 2004)
 - Inrichten werkplek
 - Basis cursus Linux
- Week 2 (16 / 20 febr 2004)
 - Basis cursus JAVA
 - Test en werkomgeving instellen
 - Inwerken in programma NetBeans
- Week 3 (23 / 27 febr 2004)
 - Usability onderzoek soortgelijke websites
 - Brainstormen structuur website
 - Opstellen Plan van Aanpak
 - Verifiëren Plan van Aanpak

Analysefase: duur 2 week

- Week 4 (1 / 5 mrt 2004)
 - Informatie vergaren
 - Gebruikers, personas definiëren
 - Systeemeisen definiëren
 - Structuur prototypes bepalen
- Week 5 (8 / 12 mrt 2004)
 - Definiëren Style Guide
 - Verifiëren definitiestudie met opdrachtgever

Conceptfase: duur 3 weken

- Week 6 (15 / 19 mrt 2004)
 - Modelleren taakdiagrammen
 - Modelleren UML-klassendiagram
- Week 7 (22 / 26 mrt 2004)
 - Modelleren UML-UseCasediagram
 - Modelleren UML-componentendiagram
 - Definiëren Style-Guide
- Week 8 (29 mrt / 2 apr 2004)
 - Prototype GUI
 - Verifiëren ontwikkelplannen met opdrachtgever

Ontwikkelingfase: duur 7 weken

- Week 9 tot en met 15 (5 apr / 17 mei 2004)
 - Opzetten database
 - Programmeren site volgens de programmeerstijl richtlijnen van het bedrijf
 - Verifiëren geprogrammeerde onderdelen met de opdrachtgever

Test en Implementatiefase: duur 3 weken

- Week 16 (24 / 28 mei 2004)
 - Bug testing
 - Usability testen aan de hand van personas
- Week 17 (31 / 4 juni 2004)
 - Eventuele verbeteringen aanbrengen aan de hand van de twee bovenstaande tests.
- Week 18 (7 / 11 juni 2004)
 - Samenstellen systeemdokumentatie

Bijlage D: Systeemeisen

Systeemeis	Pilot 2 Structuur	Pilot 3 webpagina's	Prioriteit
De beheerder moet advertentie groep kenmerken kunnen wijzigen en verwijderen	X	X	Basis
Bij het invoeren van de advertentie moet er een mogelijkheid zijn om een voorbeeld van de advertentie te bekijken	X	X	Basis
Een geregistreerde moet reactie kunnen sturen naar een adverteerder	X	X	Basis
Een geregistreerde moet zijn gegevens kunnen wijzigen	X	X	Basis
Een adverteerder moet een reactie kunnen ontvangen op zijn advertentie via email	X		Basis
De bezoeker moet op twee manieren kunnen zoeken naar advertenties, op een simpele en uitgebreide manier	X	X	Basis
De adverteerder moet bij de advertentie een of meerdere foto's kunnen toevoegen	X	X	Basis
Een geregistreerde heeft de mogelijkheid om eventueel een gebruikersnaam op te geven om mee in te loggen	X	X	Basis
De adverteerder moet zonder hulp een advertentie kunnen plaatsen		X	Basis
Een geregistreerde moet kunnen bieden op advertenties waar dat ingesteld is door de adverteerder	X	X	Basis
Een adverteerder moet biedingen kunnen bekijken voor zijn advertenties	X	X	Basis
De adverteerder moet ingelogd zijn om een advertentie te plaatsen	X		Basis
Het plaatsen van een advertentie moet bevestigd worden door de adverteerder	X	X	Basis
De website moet onder Mozilla, Internet Explorer, Opera en Netscape dezelfde functionaliteit hebben		X	Basis
De website moet gebruik maken van Cascading Style Sheets (CSS) om de stijl van knoppen, context en tabellen te definiëren		X	Basis
De website mag maar 1 soort lettertype gebruiken		X	Basis

De buitenkant van de website (GUI) dient losgekoppeld te worden van de broncode die de acties met de database uitvoert	X	X	Basis
De advertenties moeten op verschillende manieren gesorteerd kunnen worden	X	X	Basis
Wijzigingen in advertenties moeten bijgehouden kunnen worden	X		Basis
Wijzigingen in gegevens van een relatie moeten ingezien kunnen worden door de beheerder	X	X	Basis
De server mag niet langer dan anderhalve seconde de tijd nemen om een request af te handelen	X		Basis
De beheerder moet groepen kunnen definiëren met bijbehorende bedragen om een advertentie te mogen plaatsen, kosten om een afbeelding toe te voegen en je gegevens in beeld te brengen	X	X	Basis
De beheerder moet kenmerken kunnen toekennen aan advertentiegroepen	X	X	Basis
De website is alleen in de Nederlandse taal		X	Basis
Elke geregistreerde moet zijn prepaid saldo kunnen opwaarderen	X	X	Basis
De adverteerder moet zijn advertenties kunnen verwijderen	X	X	Basis
De beheerder moet de bedragen die betaald moeten worden kunnen bepalen en wijzigen	X	X	Basis
De beheerder moet wijzigingen van advertenties kunnen achterhalen	X	X	Comfort
De beheerder moet reacties kunnen inzien van bezoekers voor foute advertenties	X	X	Comfort
De beheerder moet statistieken kunnen inzien met betrekking op de hits van advertenties op de site	X	X	Comfort
Een bezoeker moet advertenties kunnen zoeken met een maximale aantal km van een postcode	X	X	Comfort
Een geregistreerde moet een lijst met favoriete advertenties bij kunnen houden	X	X	Comfort
De bezoeker moet een reactie kunnen sturen naar de beheerder bij een foute advertentie	X	X	Comfort
Een bedrijf moet reclame content gevoelig kunnen plaatsen	X	X	Comfort
De beheerder moet bedrijven goed kunnen keuren voordat ze reclame kunnen plaatsen op de site	X	X	Comfort

De beheerder moet statistieken kunnen inzien met betrekken op de inkomsten van de website	X	X	Luxe
Een bedrijf moet reclame op de site kunnen zetten	X	X	Luxe
De geregistreerde moet een bericht kunnen krijgen als er een advertentie is geplaatst die voldoet aan bepaalde criteria	X	X	Luxe

Bijlage E: Onderzoek aanvaardbaarheid systeemeis

Adverteerders zoeken naar nieuwe wegen

Het aantal on-line advertentie-campagnes is in 2002 gestegen met ruim 30% tot in totaal 10.000. De bestedingen zijn echter iets gedaald ten gevolge van de structureel lager liggende advertentietarieven.

Trendmatig is te zien dat het internet steeds vaker ook voor branding wordt ingezet in plaats van alleen voor het genereren van directe traffic naar de site. Het blijkt dat mensen banner uitingen steeds meer in hun achterhoofd houden. Ongeveer een half procent van de mensen die een banner heeft gezien, komt zonder direct te klikken, later toch bij het bedrijf uit.

Bij campagnes wordt ook steeds meer gekozen voor grote banner formaten en rich media banners. Dit is niet voor niets want rich-media banners blijken succesvol.

Doorklikpercentages van rich-media banners komen uit op 2,5% (gegevens DoubleClick). Dit terwijl de traditionele banners niet verder komen dan 0,4%.

Targetting op basis van keywords (bijvoorbeeld bij zoekmachines) en doelgroepen (bijvoorbeeld bij op doelgroepgerichte sites) wordt ook als effectieve methode gezien. In totaal werd aan het einde van 2002 42% van de campagnes op één van deze twee wijzen getarget.

Bron: <http://www.markvanwerven.nl/>

Sterke groei in online adverteren verwacht

Onderzoeksbureau Forrester en Interactive Advertising Bureau (IAB) Europa voorspellen een groei in de uitgaven voor online adverteren en interactieve marketing. Zij voorspellen dat in 2007 €5.023 miljoen besteedt wordt aan deze vorm van reclame; in 2002 was dat nog €787 miljoen.

Uit onderzoek van Forrester in twaalf Europese landen blijkt dat 74 procent van de internetters in 2002 wel eens op een online advertentie heeft geklikt. Voor 51 procent was dat minder dan een maand geleden; een stijging van 9 procent ten opzichte van het jaar daarvoor. Meer dan een kwart van deze groep ging ook over tot aankoop op de site. Een jaar eerder was dat nog 14 procent. De internetpenetratie ligt in Nederland op 71 procent. Daarmee behoort Nederland tot de top drie van landen met een hoge internetpenetratie.

Adverteerders zoeken naar nieuwe wegen op internet

Het aantal on-line advertentie-campagnes is in 2002 gestegen met ruim 30% tot in totaal 10.000. De bestedingen zijn echter iets gedaald ten gevolge van de structureel lager liggende advertentietarieven.

Trendmatig is te zien dat het internet steeds vaker ook voor branding wordt ingezet in plaats van alleen voor het genereren van directe traffic naar de site. Het blijkt dat mensen banner uitingen steeds meer in hun achterhoofd houden. Ongeveer een half procent van de mensen die een banner heeft gezien, komt zonder direct te klikken, later toch bij het bedrijf uit.

Bij campagnes wordt ook steeds meer gekozen voor grote banner formaten en rich

media banners. Dit is niet voor niets want rich-media banners blijken succesvol. Doorklikpercentages komen uit op 2,5% (gegevens DoubleClick) terwijl de traditionele banners niet verder komen dan 0,4%. Targetting op basis van keywords (bijvoorbeeld bij zoekmachines) en doelgroepen (bijvoorbeeld bij op doelgroepgerichte sites) wordt ook als effectieve methode gezien. In totaal werd aan het einde van 2002 42% van de campagnes op één van deze twee wijzen getarget.

Donderdag 6 februari 2003

Bron: www.websidestory.com

http://www.trendbox.nl/eye/news/index.html?html/040212_popups.html Amsterdam,
12 februari 2004

Bijlage F: Representative qualities

De 12 Representative Qualities zijn :

1. Correctness, Reliability, Robustness
2. Performance
3. User Friendliness
4. Verifiability
5. Maintainability
6. Reusability
7. Portability
8. Understandability
9. Interoperability
10. Productivity
11. Timeliness
12. Visibility

Correctness, Reliability, Robustness

a. Correctness

Een programma is functioneel correct als het zich gedraagt volgens de specificaties van de functies waarin het dient te voorzien. Voor de website betekent dit dat het programma de functies die het aangeeft ook goed uitvoert.

b. Reliability

Er zijn twee definities voor 'reliability' mogelijk:

- Informeel: Software is betrouwbaar ('reliable') als de gebruiker ervan op aan kan;
- Formeel: Hier definieert men 'reliability' in statistische termen – de kans dat software zal opereren als verwacht over een specifieke tijdsspanne.

De website moet betrouwbaar zijn. Iemand die op de site een advertentie wil plaatsen zal het niet waarderen als de website niet goed reageert op de invoer van de advertentie of dat deze op een totaal andere plek komt te staan.

c. Robustness

De website is 'robust' als het zich 'redelijk' gedraagt (bijv. *Niet* crasht) onder onverwachte omstandigheden. Correctness, reliability en robustness kunnen ook van toepassing zijn op het softwareproces. Hier is weer de lage drempel van overstappen naar de concurrent aan de orde. De gebruiker is er niet van gediend als het systeem crasht of behoorlijk vertraagd als hij bijvoorbeeld iets verkeerd heeft ingevoerd.

Performance

Ieder product hoort een bepaald performance niveau te bereiken. Met betrekking tot de advertentiewebsite stellen we performance vaak gelijk aan efficiency. De website is efficiënt als het zuinig gebruik maakt van de hardware.

Performance is belangrijk omdat het de bruikbaarheid van het systeem beïnvloedt.

3 factoren zijn van belang:

- snelheid (van de software);
- gebruikte opslagruimte;
- gebruikt werkgeheugen.

Performance kan op meerdere manieren getest worden:

- efficiency meten door de complexiteit van de algoritmes te analyseren;
- performance evaluation waarvoor 3 benaderingen mogelijk zijn:
 - meten (bijv. verwerkte hoeveelheid dat per tijdseenheid);
 - analyseren (van een model van de software);
 - simuleren (van de software gebruik makend van een model).

Ook performance is van toepassing op het proces; we spreken in dit geval van *productivity*.

User Friendliness

Indemarkt.info is 'user friendly' (gebruiksvriendelijk) als zijn (menselijke) gebruiker het eenvoudig te bedienen vindt.

User friendliness wordt bepaald door:

- De Grafical User Interface (GUI): de interface is overzichtelijk en de plaatsing van de functies (buttons e.d.) is logisch en overzichtelijk.
- De functionaliteit die de gebruiker tegemoetkomt om hun doelen te bereiken.

Verifiability

Indemarkt.info is 'verifiable' als zijn eigenschappen eenvoudig kunnen worden getoetst. Dit kan gebeuren door middel van:

- formele analysemethoden;
- testen.

Dit gebeurde in dit geval door de testen die in de planning worden beschreven.

Maintainability

De term 'software maintenance' wordt gebruikt om te verwijzen naar de aanpassingen die aan de website kunnen worden aangebracht nadat deze in gebruik is genomen.

Er zijn drie categorieën van software maintenance:

- corrective maintenance = het verwijderen van fouten uit de software;

Bij het gerealiseerde product is dit mogelijk door de structuur van de website en programmeer standaarden die opgelegd zijn door Damman Software.

Fouten en of wijzigingen kunnen zo moeiteloos worden verbeterd.

- adaptive maintenance = de applicatie aanpassen aan veranderingen in de omgeving (hardware of software);

Als we gaan kijken naar de Hardware zijn zal er bij snellere componenten winst worden behaald in het verwerken van processen, zoals het verwerken van database query's. Er hoeft niets veranderd te worden aan de website zelf.

Als we kijken naar de wijzigingen in de software moet er gekeken worden of de website afhankelijk is van deze software. De website draait bijvoorbeeld op Apache-Tomcat. Zal deze versie worden geupdate of er wordt gekozen voor een ander alternatief dan zal er vooraf moeten worden bepaald hoe de website overgezet moet worden. Dit kan per programma of versie verschillen.

- perfective maintenance = de software aanpassen om zijn kwaliteiten te verbeteren.

Aangezien het Indemarkt.info wordt gebouwd volgens een structuur waarmee het programmeer gedeelte is losgekoppeld van de Grafical User Interface kunnen hier vrij gemakkelijk wijzigingen in aangebracht worden. Hierdoor is het ook mogelijk losse componenten individueel te optimaliseren.

Maintainability omvat twee verschillende taken:

- Repairability: de website Indemarkt.info is 'repairable' als zijn fouten met een relatief geringe inspanning kunnen worden aangepast. Repairability beïnvloedt de reliability (betrouwbaarheid) van de software. Door broncodes op te splitsen in klassen, opmaakpagina's en servlets is het mogelijk om alle onderdelen snel te verbeteren en/of uit te breiden.
- Evolvability: De mogelijkheid om software in verloop van tijd van nieuwe

functies te voorzien of bestaande functies aan te passen. Hoe vaker dit gebeurt, hoe groter de kans op fouten echter wordt en hoe minder de evolvability. Evolvability is ook van toepassing op het proces.

Reusability

Resuablility lijkt erg op evolvability. Bij product evolutie groeit het product naar een nieuwe versie toe. Bij het hergebruiken (reusability) wordt gebruik gemaakt van onderdelen uit het ene product om een ander product te bouwen.

Het is belangrijk tijdens het ontwikkelen van het component al rekening te houden met reusability. Bij Indemarkt.info staat dit op een hoog niveau omdat het bestaat uit de losgekoppelde onderdelen. Deze kunnen dus makkelijk worden aangepast/toegevoegd of hergebruikt om weer een nieuwe functie te vervullen.

Portability

Indemarkt.info is portable (overdraagbaar) wanneer het functioneert in verschillende omgevingen zoals verschillende hardware platformen of besturingssystemen. Dankzij het gebruik van Java Servlets en Java Server Pages is dit mogelijk. De website draait op Apache Tomcat versie 5. Hierdoor kan je de website op zowel Unix als Windows laten draaien.

Understandability

Understandability (begrijpelijkheid) gaat over de interne complexiteit van de website. De interne kwaliteit is hoog wanneer het ontwerp (bijv. broncode) zo begrijpelijk mogelijk is opgezet.

Alle onderdelen zijn opgezet volgens dezelfde structuur en zal mede hierdoor al snel een vertrouwd gevoel geven. Door gebruik te maken van de programmeerstandaarden die worden gevoerd door Damman Software zal de broncode makkelijk te begrijpen zijn.

Externe understandability is een onderdeel van user friendliness (gebruikersvriendelijkheid)

Interoperability

Interoperability betekent samenwerkingsvermogen. Het gaat hier om de samenwerking tussen verschillende softwareproducten (zoals verschillende databases).

Samenwerkingsvermogen betekent ook dat de verschillende softwareproducten elkaar niet storen.

De advertentiesite heeft te maken met een databaseserver, webserver en een

programmeertaal die met elkaar samen moeten werken. In dit geval is de webserver speciaal getest voor Java en is er een driver geschikt gemaakt die een connectie kan maken met de database.

Productivity

Met productiviteit wordt het verwerkingsvermogen van de website bedoeld. Het meet zich met de efficiëntie van afhandelen van de acties die door de bezoeker wordt uitgevoerd. Een efficiënt proces zorgt voor een snellere verwerking van deze acties.

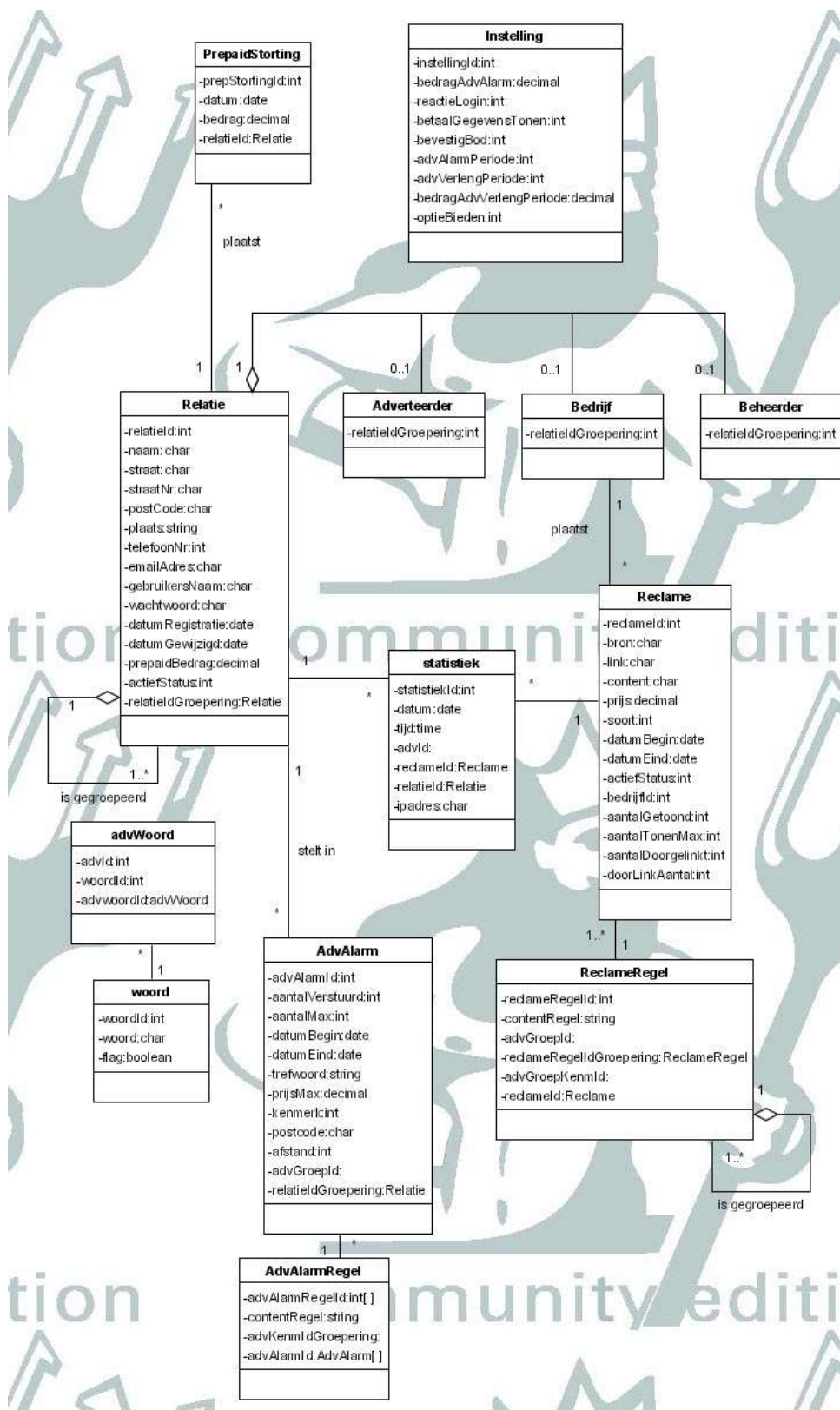
Timeliness

Timeliness is een proces-gerelateerd kwaliteitsaspect welke gaat over het op tijd opleveren van de website. De klant wil altijd redelijk snel resultaat zien. Daarom is het van belang een realistische planning te maken en daar zoveel mogelijk aan te houden.

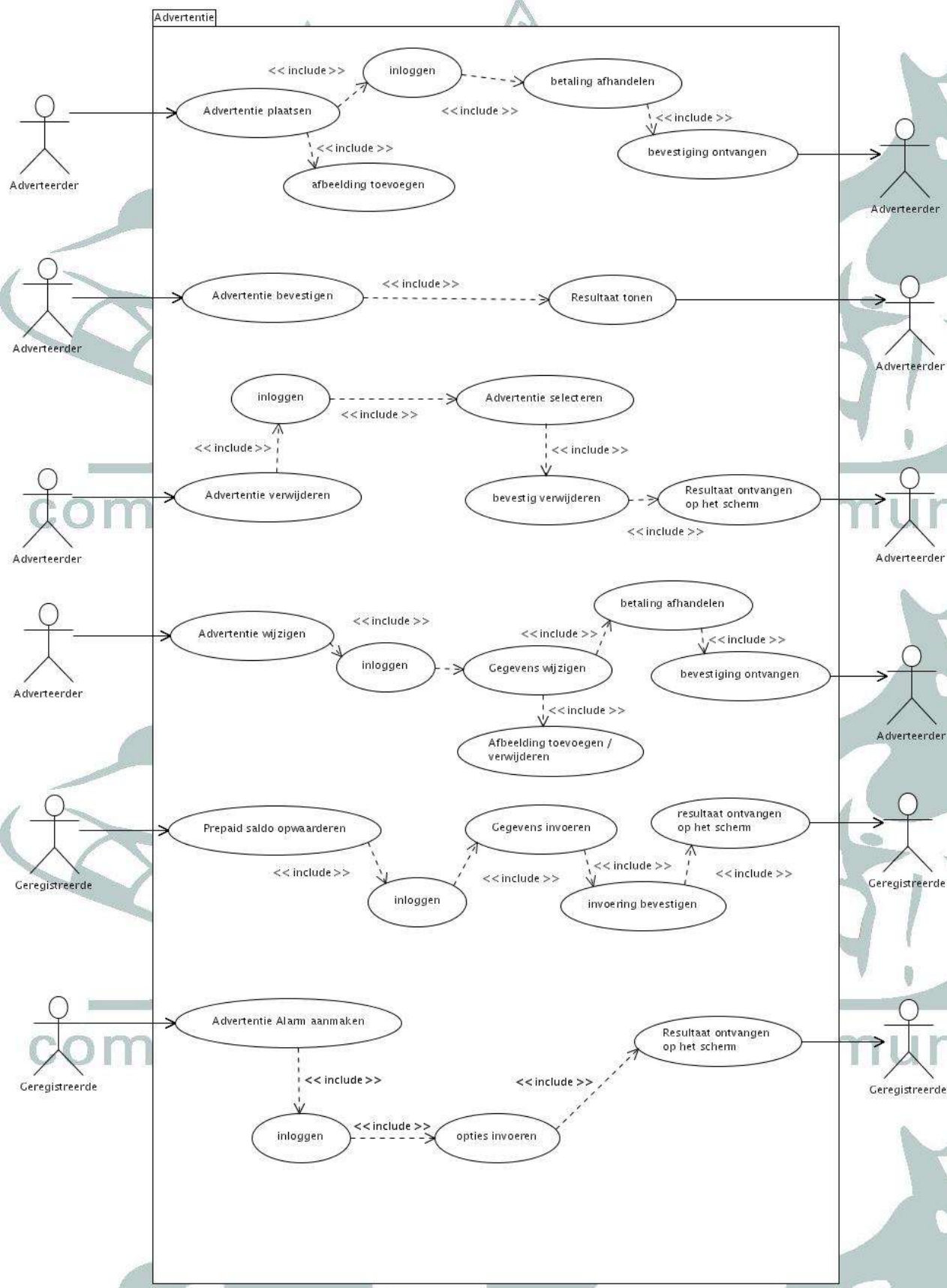
Visibility

Visibility betekent zichtbaarheid en is gebaseerd op het idee dat tijdens de ontwikkeling voldoende inzage gegeven moet worden door de ontwikkelaars. De verschillende stappen en de status van het ontwikkeltraject worden goed gedocumenteerd.

Bijlage G: UML Klassendiagram



Bijlage H: UML Use-Case diagram



Bijlage I: Planning uitwerken van pilots

Planning uitwerking Pilots			
Week	Activiteit		Producten
<p>9</p> <p>5 t/m 9 april 2004</p>	<ul style="list-style-type: none"> Pilot 1 (database) Basisstructuur Website opzetten Opzetten Database 	<p>Structuur aanmaken volgens Model View Controller</p> <p>Database aanmaken tabellen en attributen invoeren testgegevens aanmaken javabean aanmaken voor connectie met database</p>	Pilot 1
<p>10</p> <p>12 t/m 16 april 2004</p>	<ul style="list-style-type: none"> Pilot 2 (model / controller) Uitwerken gedeelte beheerder 	<p>Programmeren:</p> <p>inloggen / uitloggen en hoofdscherm invoeren categorieën invoeren kenmerken per categorie wijzigen van standaard instellingen updaten saldo gebruiker</p>	
<p>11</p> <p>19 t/m 23 april 2004</p>	<p>Uitwerken gedeelte bezoeker</p>	<p>Programmeren van:</p> <p>zoeken naar advertenties geregistreerde aanmaken advertentiedetails ophalen advertenties ophalen per categorie</p>	

Planning uitwerking Pilots			
12 26 t/m 30 april	Uitwerken gedeelte geregisteerde	Programmeren van : plaatsen van een advertentie wijzigen van een advertentie verwijderen van een advertentie bieden op een advertentie bevestiging versturen advertentie bevestigen bod bevestigen advertentie alarm aanmaken advertentie alarm verwijderen advertentie reactie versturen prepaid betaling verwerken afbeelding toevoegen afbeelding verwijderen	Pilot 2
13 3 t/m 7 mei			
14 10 t/m 14 mei	Pilot 3 (view) uitwerken van schermen voor bezoeker uitwerken schermen voor	Uitwerken van het hoofdscherm met frames. Titel, intro, Uitwerken van de volgende schermen: zoeken zoekresultaat advplaatsen advdetail registratie inloggen	

Planning uitwerking Pilots			
15 17 t/m 21 mei	Uitwerken schermen voor adverteerder / geregistreerde	overzicht advwijzigen advverwijderen actieresultaat advalarmmaken advalarmverwijderen	Pilot 3

Bijlage J: Programeerrichtlijnen

Naamgeving variabelen

Om de leesbaarheid van de code te verbeteren worden variabelen afhankelijk van hun type voorzien van een kleine letters prefix gevolgd door de eigenlijke naam beginnend met een hoofdletter. De basistypen hebben een meestal eenletterige prefix:

```
int: i of i32
short: i16
long: i64
String: s
char: c
float: f
double: f
boolean: b
```

Bij het gebruik van objecten wordt zeker tot 8 tekens de hele objectnaam gebruikt. Daarna kan de typenaam worden ingekort door b.v. klinkers weg te laten. In geval van twijfel is de kortste oplossing de beste oplossing.

Voorbeelden:

```
int iHelp;
boolean bHelp;
Object objectHelp;
```

Array variabelen worden voorafgegaan door een a en een cijfer die de dimensies aangeeft. In geval van een 1 wordt deze weggelaten:

```
int aiHelp[];
int a2iHelp[][];
```

Ook worden bij arrays geen meervouden gemaakt omdat dit al aan de prefix te zien is: dus Auto aauto[]; ipv Auto aautos[];

Klassevariabelen (members) worden voorafgegaan door een underscore:

```
private int _aiHelp[];
```

Statische (klasse)variabelen worden voorafgegaan door een s:

```
private static int s_aiHelp[];
```

Naamgeving methoden

Methoden beginnen met een kleine letter en worden net zoals de variabelen verder in de Hongaarse methode geschreven:

```
public void lichtSchakelen(boolean bAan)
public int bepaalGrootte(Object objectIn)
```

Gebruik accolades:

Altijd symmetrisch dat heet op een nieuwe regel:

```
if (bWaar)
{
    bWaar = false;
}
```